

AD-A069 868

NAVAL INTELLIGENCE PROCESSING SYSTEMS SUPPORT ACTIVITIES--ETC F/6 9/2  
INTEGRATED DATABASE DEVELOPMENT AND DESIGN GUIDE. VERSION 2.0.(U)  
MAY 79

UNCLASSIFIED

NIPSSA-UM-8000/2.0-5/79

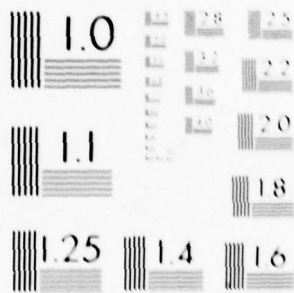
NL

1 OF 7

AD  
A069868







LEVEL

2

14 NIPSSA-  
UM-8000/2.0-5/79

AD A 069868

6

INTEGRATED DATABASE  
DEVELOPMENT AND DESIGN GUIDE  
Version 2.0

DDC  
RECEIVED  
JUN 13 1979  
C

NAVAL INTELLIGENCE PROCESSING SYSTEM  
SUPPORT ACTIVITY  
(NIPSSA)

THIS DOCUMENT IS BEST QUALITY PHOTOGRAPH  
THE COPY FURNISHED TO DDC CONTAINS A  
SUFFICIENT NUMBER OF PAGES TO BE  
REPRODUCED EXACTLY.

This document has been approved  
for public release and sale; its  
distribution is unlimited.

VERSION 2.0

11 MAY 1979

DDC FILE COPY

12 592 p.

PREPARED BY  
DATA ADMINISTRATION DIVISION

79 06 12 164 410 508 Am

## **DISCLAIMER NOTICE**

**THIS DOCUMENT IS BEST QUALITY  
PRACTICABLE. THE COPY FURNISHED  
TO DDC CONTAINED A SIGNIFICANT  
NUMBER OF PAGES WHICH DO NOT  
REPRODUCE LEGIBLY.**

#### ACKNOWLEDGMENT

The author wishes to express his grateful appreciation to the NIPSSA personnel who assisted in the preparation of this Guide. Their comments, critiques, and proofing of the material contributed significantly to the preparation of a useful tool for integrated database design. Special thanks go to Mrs. Janelle Loper, Mrs. Kitty Mears, Mr. Ray Werner, Mr. Jack Filigenzi, and Mr. Bill Potter for their critical review.

Accession for	
NTIS - GPO	
DOI - 100	
Unpublished	
Justified	
Form 50 per file	
By	
Date	
Availability	
Dist	Avail and/or special
11	12/12

79 06 12 164

## TABLE OF CONTENTS

1.	INTRODUCTION
1.1.	Background
1.2.	What Is A Database?
1.3.	Why Use A database?
1.4.	Purpose of The Guide
1.5.	Scope of The Guide
1.6.	The "Universal" Database (UDB) Concept
1.7.	Data As A Resource
2.	PROCEDURAL OVERVIEW
2.1.	Integrated Data Dictionary (IDD) Facility
2.2.	Design and Development Methodology
2.3.	Phase I. The Initial Requirement Definition
2.4.	Phase II. Database Foundation Design
2.5.	Phase III. Database Integration
2.6.	Phase IV. Batch Input Processing and File Conversion Development
2.7.	Phase V. Batch Reporting from The Database
2.8.	Phase VI. On-line Input Processing Capability Development
2.9.	Phase VII. On-line Reporting from the Database
2.10.	Summary of Design and Development Milestones
2.11.	How to Use The Guide
3.	DETAIL DESIGN AND DEVELOPMENT INSTRUCTIONS
3.1.	Phase I. Initial Requirement Definition
3.2.	Phase II. Database Foundation Design
3.3.	Phase III. Database Integration
3.4.	Phase IV. Batch Input Processing and File Conversion Development
3.5.	Phase V. Batch Reporting from The Database
3.6.	Phase VI. On-line Input Processing Capability Development
3.7.	Phase VII. On-line Reporting from The Database



## APPENDICES

### A. DESIGN INFORMATION PREPARATION INSTRUCTIONS

- A.1. General Instructions
- A.2. Program Support Library (PSL) Use
- A.3. New IDD User Entries
- A.4. IDD Subsystem Definition Entry
- A.5. IDD User/Subsystem Association Entry
- A.6. IDD Data Element Definition Entry
- A.7. IDD Group Data Element Definition Entry
- A.8. IDD Entity Data Definition Entry
- A.9. IDD User Database Authorization
- A.10. Service Analysis Preparation
- A.11. IDD Report Program Entry
- A.12. Batch Input Processing (IP) Specification  
Definition
- A.13. (tba)
- A.14. On-line Input Processing (IP) Definitions
- A.15. (tba)
- A.16. (tba)
- A.17. Users Guide Preparation Instructions

### B. CONVENTIONS

- B.1. Data Element Name Conventions, General
- B.2. Common Data Element Name Prefixes
- B.3. Data Element PICTURE Definition Conventions
- B.4. Data Element USAGE Conventions
- B.5. Initial VALUE Conventions
- B.6. IP Naming Conventions
- B.7. COBOL Programming Conventions
- B.8. Entity Relationship Template
- B.9. General UDB Record Types
- B.10. PSL Input Tape Format Standards
- B.11. CRT Function Key Use

### C. STANDARD RECORD STRUCTURES

- C.1. Single Owner/Member Record Entity Relationship
- C.2. DBOMP (Record Owns Self) Relationship

### D. DOCUMENTATION STANDARD FOR ADP SUBSYSTEMS

E.	CODES AND TABLES
E.1.	Security Classifications
E.2.	Data Element Standards
E.3.	Language Statement Names
E.4.	Standard Output Forms
E.5.	IP Function Options
E.6.	DECODE Table
E.7.	PRESENCE Table
E.8.	Validation Test Identifier Tables
E.9.	DATA-SUBJECT Table
E.10.	Element Designators
E.11.	Predefined Data Elements
E.11A.	Predefined Data Elements (IDD Syntax)
E.12.	Frequency of Service Codes
E.13.	Priority of Service Codes
E.14.	Response Time of Service Codes
E.15.	Service History Codes
E.16.	Service Output Modes
E.17.	Standard TP-IP Function Command Codes
F.	SAMPLE USERS GUIDE
G.	IDD SYNTAX (Separate Cover)
H.	SAMPLE DATABASE DEFINITION
I.	GLOSSARY OF TERMS
J.	JOB CONTROL LANGUAGE (JCL)
J.1.	PSL Update
J.2.	Batch IP Compilation
J.3.	CULPRIT Reports
J.4.	IP Execution
K.	DEVELOPMENT SCHEDULE AND DELIVERABLES
L.	UNIVERSAL DATABASE (UDB) TECHNICAL APPROACH (Separate Cover)



## 1. INTRODUCTION.

### 1.1. Background.

↘ The NIPSSA Integrated Database Development and Design Guide is the result of several years of experience developing integrated database applications. It brings together into a structured methodology techniques which have survived trial by implementation. Some of the procedures within the Guide are relatively new and may require additional clarification. Comments and recommendations are welcomed and should be addressed to the NIPSSA-3DN Database Administrator (DBA). 6 PJ  
13

### 1.2. What is A Database?

James Martin, one of the best known authors on data processing, defines database in his book, COMPUTER DATA BASE ORGANIZATION: "A database may be defined as a collection of inter-related data stored together with as little redundancy as possible to serve one or more applications in an optimal fashion....".

The definition becomes more meaningful when it is analyzed in sections.

"... a collection of inter-related data..." means that data, such as date, time, and place of birth, which are dependent upon each other to form a picture, are collected together. One of the most important tasks of designing a database is giving the user access to related data.

"... stored together..." means that this related data is placed on some type of storage medium in such a way that it is consistently and conveniently controlled as a single entity. This permits gathering needed information from a single location rather than having to capture it from multiple locations.

"... with as little redundancy as possible ..." points to a problem which always exists when data is kept in different places. If a manager keeps a personal file of tasks in progress s/he is unlikely to omit the subject and date received, for example, just because that information is

also kept by someone else. This redundant information costs money. It requires someone to copy it from another source, space in which to keep it, and time to check back to see if the original source has changed the data since the copy was made. It also leads to confusion when the original data is changed and the copy is not. A database, properly designed, can eliminate most redundant data and more effectively control that which remains.

"... to service one or more applications ..." describes one of the most useful, yet controversial, features of a well-designed database. It is desirable for a single repository to support, concurrently, a personnel file; a payroll file; an organization structure definition; and a project management capability. What is often difficult is to get the users of each of these facilities to share the data which is common to all.

"... in an optimal fashion ..." relates to one of the greatest potentials of a database. Because of its organized approach to processing data, it is possible to develop common procedures and computer programs to handle all of the data in the database. Using repetitive procedures reduces the cost of each application just as an assembly line reduces the unit cost of building an automobile. This technique also reduces the possibility of error in handling data, increasing the reliability and usefulness of the data to the end user.

### 1.3. Why Use A Database?

As the world in which we live and work becomes more complex, daily decisions depend more and more on the availability of accurate and up-to-date information.

The ability of data processing organizations to keep pace with user's needs for information is increasingly taxed. As the work becomes more complex, so do the relationships between the data which describes the work. And quickly the number of skilled personnel required to keep up with the demand exceeds limits imposed by funding and other restrictions. A way of overcoming these limitations is finding a better approach to process data. The database management system (DBMS) and the computer programs which support its database, is a better way.

The bottom line is that a database can give users of information a new tool to expand and improve their capability to meet today's information requirements.

#### 1.4. Purpose Of The Guide.

The development of an integrated database is an expensive and highly detailed project. The speed with which applications can be added or enhanced is directly proportional to the analysis resources available. The most time-consuming part of the analysis is the definition of the data elements and their relationships. Once this is done the remainder of the design effort falls rapidly into place. The Guide provides a step-by-step set of instructions which lead to a subsystem implementation of the user's desired application. At the same time it will permit the user, who knows more about the data than anyone else, to perform the initial phases of the analysis.

The Guide is meant to provide the complete picture and steps required to implement a database application. For this reason, all of the procedures to be followed by both user and Data Administration (DA) personnel are included. Section 2 identifies who will perform each step of the development sequence.

Section 2 of the Guide provides a non-ADP user orientation of the design process. Section 3 expands the description for use by ADP-oriented analysis personnel.

The commitment to developing an integrated database is one with far-reaching impacts. We can no longer afford to develop software in the piecemeal and independent world which has characterized file-mode processing. It is essential that each application developed for the integrated database consistently follow the same path to implementation. There is very little room for independence if an effective database is to be the end result.

This environment is particularly difficult to achieve when multiple user groups, each with ADP support or services, undertakes to develop database applications. Without some consistent guidance, important parts of an application may be omitted. Worse yet, some applications may actually interfere with others.

The situation becomes more acute when development is performed by outside contractors or consultants. Each software firm has their own approaches and techniques for developing software. While this arrangement is often satisfactory in a file-mode environment, it will lead to chaos in the database arena.

The Guide attempts to solve most of these approach problems by defining a consistent technique for development which can be easily followed by all developers of database applications.

It must be understood, however, that the Guide is only as good as the DA staff who administers and enforces its conventions. Success or failure is not dependent upon the level of detail of instruction within the Guide but upon the consistent application of the principles it supports.

#### 1.5. Scope Of The Guide.

The Guide begins with the initial definition of an application requirement by the potential user. The user, as discussed throughout the Guide, is the end recipient of ADP services and ADP analysis personnel who serve as the user's agents. The Guide then proceeds through several analysis and development steps, including periodic review and decision points, to the completion of the development and implementation of the application.

Hardware support for the application is not specifically addressed by the Guide. Where hardware must be procured to support the application, validation and procurement approval must be performed in parallel with the preparation of software specifications. This hardware requirement must be documented as part of the subsystem specification.

The Guide does not include a number of common features of the integrated database which are provided independent of application subsystems. These features include recovery of databases during processing malfunctions, logging of transactions, user access protection, and security control. Detailed explanations of these common features and their relationship to applications is available through the DA staff.



#### 1.6. The "Universal" Database (UDB) Concept.

An integrated database is typically characterized by the presence of a large number of inter-record relationships. As the scope of the database expands to meet user requirements, the complexity of these relationships increases. The time and effort required to design the database structure increases exponentially with complexity.

A simplified database structure has been developed which eliminates many of the problems associated with increased design complexity. This structure has been named the Universal Database (UDB) because it permits nearly unlimited association between database record types. This Guide will address only UDB design techniques.

Briefly, the major advantages of UDB are:

- a. Database design time is reduced by approximately 30 per cent.
- b. Updating of the database is consistently fast regardless of the size of the database or the percentage of its space utilized.
- c. Every record containing substantive data in the database may be individually accessed, increasing retrieval speed for most on-line query requirements.
- d. New relationships between record types may be added without software modification. This improves responsiveness to user needs and eliminates database restructuring.
- e. Additional data elements may be added to a record type without affecting existing database records or programs, eliminating the need to restructure the database.
- f. Multiple users of the same application are isolated from each other logically while physically sharing storage space. This reduces the cost of storage space and eliminates redundant software development.

- g. Many different applications can, if desired, share the same storage space without interfering with each other. This is particularly useful for small applications which require little storage space.

#### 1.7. Data As A Resource.

Historically, ADP has viewed data as a possession of individual users. Design of ADP systems has been oriented to the outputs, typically reports, desired by a single user. The data files rarely contained data elements which were not required for the immediate needs of the primary user. The concept of an integrated database has forced a change in this thinking.

Database encompasses entire organizations. The data stored is often used by more than one person and may be organized for many different outputs. DATA BECOMES A RESOURCE of the organization instead of a possession. The development concepts associated with this approach have moved from an output-oriented philosophy to a complete resource philosophy. This requires that the system designer significantly alter his/her thinking and look beyond the immediate benefits of a data file to the potential needs of the user organization as a whole. This Guide, with its methodology, will assist the database designer in following this approach.

## 2. PROCEDURAL OVERVIEW.

This section describes, in general terms, the steps which are to be performed to accomplish the description of a database application. This section is intended for use by user and management personnel who have a minimum of ADP knowledge. Detail step instructions for ADP personnel are presented in Section 3.

### 2.1. Integrated Data Dictionary (IDD) Facility.

It is essential that the design of a large and complex database be controlled and monitored. This monitoring is best performed using a software tool known as a data dictionary. The data dictionary, similar to a standard reference dictionary, defines and relates terms and data descriptions that are used within the database.

The integrated database uses the Integrated Data Dictionary (IDD) software package developed by Cullinane Corporation. IDD works in close association with the Integrated Database Management System (IDMS), also marketed by Cullinane. IDMS is the heart of the integrated database and performs all database support functions.

IDD is used extensively during the design and development process. Each major step of Section 3 includes entries to the data dictionary. This approach centralizes all design efforts and insures continuity with the design of the existing operational database. Reports produced by IDD permit rapid review of the design efforts and improve the accuracy of the overall process. Dictionary entries are structured so that the entire text of a subsystem specification, described in Appendix D may be prepared from the dictionary contents.

### 2.2. Design and Development Methodology.

This Guide presents a carefully structured and somewhat rigorous methodology for database design and development. The order of steps illustrated in Figure 2.1 has been developed through the experience derived from database design efforts using the previous version of this Guide.



Where multiple interrelated databases are being developed concurrently, it is essential that all developers adhere to the design approach. The end product, the operational database, *MUST* be harmonious and smoothly integrated. Therefore, the implementation and enforcement of effective design techniques is required.

The design and development process described in this Guide is broken into seven phases:

- a. Phase I defines the initial scope of the ADP support requested by the user. This phase includes a detailed definition of the end product(s) to be achieved, the sources of information required to support the application, and the relative importance of the application. Where practical, an analysis of services to be provided will be performed.

The remaining phases of the development effort are performed where a new database development or a major modification to an existing database is indicated. When the requested support is serviced by new or redefined outputs from an existing database, only Phases V and VII are performed.

- b. Phases II and III are the primary design phases. During these steps, the foundation of a database is defined and constructed. Data elements are defined and grouped. The elements and groups are integrated into the database structure to become a homogeneous part of the total database.
- c. Phases IV and V define the development of batch facilities to maintain and report from the database. Here program generating software and the CULPRIT query and report writing package perform much of the effort necessary to rapidly implement the new facility.
- d. Phases VI and VII are included when development requires an on-line interface to the database. Specialized support software provides flexible interfaces to the database for a majority of applications.

Figure 2.1 illustrates task steps as lines and milestones as circles. The tasks are performed by a combination of user and data administration personnel. Two terms are commonly used throughout the Guide which must be clearly understood:

- a. "USER" is defined as those personnel within the requesting organization or contractor personnel supporting the requesting organization's application.
- b. "DATA ADMINISTRATION (DA) STAFF" is defined as those NIPSSA personnel who are supporting database development. These include the project manager as well as database administration personnel.

#### 2.3. Phase I. The Initial Requirement Definition. (Step 1).

The scope of the requirement is described in detail. Where the application is large and/or requires numerous services, a service analysis is performed. Once the application has been defined, it is reviewed to determine its level of interface with existing database facilities.

#### 2.4. Phase II. Database Foundation Design.

This phase is the most critical part of the design process. Definition of data elements and element groups provides the basis for support of all services associated with the database. Accurate and detailed definition of all known and anticipated data elements is essential to an effective implementation of the application. The temptation to short-cut or combine steps in this phase must be resisted to insure that the foundation of the database is properly constructed.

2.4.1. Define the Individual Data Elements Required By the Application. (step 2) Each data element is separately defined as completely as possible. This step requires close cooperation between an interviewing analyst and end user personnel who understand the purpose and definition of the elements. Once defined in data dictionary format, element descriptions are stored in a programming support library (PSL) for easy manipulation. This step is performed by user personnel (in-house or contractor).

2.4.2. Review Data Element Definitions. (step 3) The DA staff reviews data element definitions for:

- a. Conformance with the data dictionary usage conventions.
- b. Consistency with established conventions and standards for data element names, formats, and content.
- c. Interaction of data elements with the established database definitions. Elements which already exist in the database are flagged for special handling later in the design process.

2.4.3. Correct Data Element Definitions and Define First Level Groups of Elements. (Step 4) The user's ADP personnel correct any element discrepancies found in the previous step and update the PSL. Elements are then associated into logical groups, such as place-of-birth and date-of-birth. Dates are themselves logical groups, composed of month, day, and year. The prepared groups are stored on the PSL following the last of the individual elements.

2.4.4. Review Element Definitions. (Step 5) The DA staff reviews the data element corrections and groups in the same manner as the individual elements were reviewed. Discrepancies are noted for user correction.

2.4.5. Correct Data Element Groups and Define Entity Relationships. (Step 6) User ADP personnel correct any discrepancies found in the previous step and update the PSL. Major entity relationships are defined among data elements and element groups. This is the final association step. Particular attention is paid to elements which have been identified as existing in the current database.

At the end of Phase II, the data which will support the requested application have been defined. It is possible, even probable, that additional elements have been defined during this phase which will prove useful to current or future applications. The ADP analysts must exercise all of their skills during Phase II to cover as much ground as possible. Long-term maintenance and support costs will be directly related to the analyst's interviewing skills.

## 2.5. Phase III. Database Integration.

This phase is performed by DA staff personnel. During the steps of Phase III, the entities developed in Phase II are integrated into the schema supporting the production database environment.

2.5.1. Define Entity Relationships to the Existing Database. (Step 7) This step identifies the logical locations to be assigned to each entity group defined in Phase II. Data elements which were flagged as redundant to the existing database are handled first. New elements which are related to the flagged elements are reviewed to determine if they should be added to the record type containing the flagged element. A pictorial representation of the entity relationships is prepared and merged with existing database pictorials.

2.5.2. The PSL is updated with any final revisions of the data definitions and the data dictionary updated. (Step 8) Dictionary reports are prepared and reviewed to be sure that overall database definition consistency has been maintained. The schema is then updated to add the new application.

At the completion of Phase III, the definition of the new application is resident in the production schema. It is now possible to begin development of supporting software.

## 2.6. Phase IV. Batch Input Processing and File Conversion Development.

This phase of the database development begins actual software creation functions. Where existing data is stored on machine-readable media, programs must be written to convert the existing data into database format. Input processing formats for batch input are defined and stored on PSL. The input processing program generator is utilized to create COBOL programs for updating the database. A users guide is prepared illustrating all input processing functions.

Finally, an operational test period begins where the user is given direct assistance for a two-week period. Once this assistance period is complete, the application is ready for batch use.

2.6.1. Define Input Processing (IP) Formats for Batch Support. (Step 9) Each record type defined within the schema



must be provided with the means for receiving data. In the batch processing environment, this means is provided through a series of independently created programs. Each program interprets punched-card data and performs validation of individual data element values, followed by defined database maintenance functions.

Batch IP formats reserve the first four positions on the punched card for identification and function definition. The remaining 76 positions are used to locate and update the database record type associated with the IP. The organization of each IP format is defined as an entity relationship using elements defined in Phase II. The resulting IP definition is stored on PSL. Once all IP formats have been defined and reviewed by the DA staff, the formats are transferred to the data dictionary.

2.6.2. Prepare IP Program Parameters for Batch Processing. (Step 10) The user's ADP analysts define the processing to be performed by each IP using a parameter language. Each parameter group is stored on the PSL. The parameters are provided to the IP generator program and a compiled COBOL program created.

2.6.3. Prepare Test Data for Batch IP's. (Step 11) The user's ADP analysts prepare a comprehensive set of test data for each IP. Test data must include correct and incorrect processing of every field in the IP. Where field combinations are interdependent, each possible combination of possible good and bad data must be tested. This test data is stored on a PSL for future use.

2.6.4. Test Batch IP's Using Prepared Test Data. (Step 12) Each IP is tested in a production environment using the test data prepared in the previous step. Where production data is available, tests are run using this data also. Where required, IP's are corrected and retested.

2.6.5. Prepare Initial Subsystem Users Guide. (Step 13) A basic users guide format is provided as part of this Guide. All database application guides will utilize the same format. In many cases, the first two sections of the guide will be nearly identical except for those areas unique to the application. A comprehensive set of instructions for preparing batch input data will be prepared, each IP being individually described.

2.6.6. Perform Operational Test of Input Processing Facility. (Step 14) Using the personnel who will prepare data for the application, test and evaluate the input processing instructions. The instructions may be altered as required to eliminate confusion. During this test period, the end user will be encouraged to exercise the system as fully as possible to locate any problems. Problems found will be evaluated and corrections made. Appropriate data dictionary entries will be made where required.

2.6.7. Define Conversion Criteria for Existing Machine-readable Data. (step 15) Where the database will replace an existing machine-readable file, it is necessary that conversion specifications be defined for each data element. This is particularly applicable where existing data values do not conform to standards adopted subsequent to the original file definition.

2.6.8. Prepare Conversion Software and Convert Existing Data. (Step 16) Software is prepared to convert the existing file data to batch IP format. Where possible, this conversion is performed utilizing CULPRIT. Converted data is passed through the batch IP programs to insure that validation criteria is consistently applied to all data within the database.

## 2.7. Phase V. Batch Reporting From The Database.

This phase may occur under two general conditions:

- a. The requested services may require only the preparation of additional output capabilities utilizing existing database capacity.
- b. The requested services are developed by an entirely new database application or a major enhancement of existing database facilities.

The process within this phase is basically the same for either condition. ADP analysts, working with end user personnel, define the outputs required. These are individually identified, defined, and documented. The data elements required to support the output are identified. Output facilities are developed and the users guide is updated to reflect the increased capability.

2.7.1. Define the Reporting Requirements. (Step 17) Each report is individually defined. The data elements required to support each reporting function are associated with the report. The report description is stored in the PSL. Once the required reports are defined, the PSL is reviewed and transferred to the data dictionary.

2.7.2. Prepare Report Capabilities. (Step 18) Each required reporting function is developed and tested. Where possible, the report function is satisfied using CULPRIT. COBOL Report Writer is permitted where the nature of the report is too complex for CULPRIT. Approval of the DA project manager is required before using facilities other than CULPRIT.

2.7.3. Update the Batch Users Guide. (Step 19) A description of each report is prepared and inserted into the appropriate section of the users guide. Each reporting function is separately described. The description must include illustrations of the report output, control parameters, and options, if any.

## 2.8. Phase VI. On-line Input Processing Capability Development.

This phase is performed when the application requires that on-line data entry be permitted. The phase includes the definition and development of on-line data entry screens, preparation of test data, testing, and documenting the screen instructions.

2.8.1. Define On-line Data Entry Requirements. (Step 20) The needs of each level of on-line user must be identified. Screen displays must be tailored to:

- a. The level of user understanding of the application system.
- b. The extent to which each person will be permitted to alter the contents of the database.

2.8.2. Prepare On-line Screen Definition Parameters. (Step 21) Prepare parameters for on-line screen program generation. These parameters will be applied to screen generating software to create COBOL processing programs operating under the teleprocessing monitor.



2.8.3. Create On-line Screen Data Entry Programs and Test. (Step 22) Create the necessary on-line screen programs to fully support on-line data entry. Prepare test entries to completely test all screens. Each data field within the screen must be tested for both good and bad data values. Where interaction between fields occurs, all combinations of good and bad data must be applied. This data will be stored in a PSL for future testing.

2.8.4. Update the Users Guide to Support On-line Data Entry. (Step 23) Each on-line data entry screen will be individually documented in a section of the users guide reserved for on-line use. Each screen will be illustrated and all functions fully explained. Interaction with other screens through function or special controls will be explained. Where several screens interact, a pictorial diagram showing the relationships between screens will be provided to assist users in fully utilizing the facilities.

2.8.5. On-line Facility Operational Test. (Step 27) Provide direct support to on-line users. Review facility documentation for consistency and clarity. Update as required.

## 2.9. Phase VII. On-line reporting From the Database.

This phase is included when development requires that special on-line reports be developed which cannot be adequately supported through use of available generalized query facilities. Where available generalized query facilities are used to prepare repetitive reports, each of those reports and any variable options are individually defined.

2.9.1. Define On-line Reporting Requirements. (Step 24) Prepare detailed descriptions of the on-line report functions which must be prepared to support the user's requirements. Identify the data elements which are required to effect each reporting function.

2.9.2. Prepare reports using generalized query facilities where possible. (Step 25) Where not practical, prepare specifications for report program creation. Create reports and test.

2.9.3. Update the Users Guide for On-line Reporting. (Step 26) Prepare instruction entries for each on-line reporting

function developed. Include illustrations of the report display. Identify optional features. Where multiple screens or function options are present, describe each. Provide a pictorial of the interrelation between multiple screens.

#### 2.10. Summary of Design Milestones.

2.10.1. A- Initial service request has been received from the user.

2.10.2. B- The application definition and purpose have been established. Determination has been made that the request can be serviced by:

1. An existing database capability,
2. Modification of an existing database capability,  
or
3. Development of a new database capability.

2.10.3. C- Individual data elements have been defined in IDD format. The element definitions have been entered in a PSL defined by the DA staff. A complete list of elements has been delivered to the project manager by the user for review.

2.10.4. D- The project manager and database administrator have reviewed the data element definitions for corrections and adherence to standards and conventions. A determination of the scope of the development has been prepared.

2.10.5. E- Data elements have been grouped according to their interrelationships.

2.10.6. F- The DA staff has reviewed the initial data element groupings for consistency and made any necessary adjustments.

2.10.7. G- Database entities have been defined. Entry points into the existing database have been identified. IDD entries have been prepared.

2.10.8. H- Entity relationships have been defined. Redundant elements have been resolved.

2.10.9. I- All logical records in the proposed database have

been identified. The IDMS UDB database definition (schema) has been updated by the DBA.

2.10.10. J- Batch IP formats have been defined and stored in the data dictionary.

2.10.11. K- Batch IP program parameters have been prepared and stored on PSL.

2.10.12. M- Batch IP's have been successfully tested. Errors have been corrected and the PSL reflects operational IP parameters.

2.10.13. N- The Batch facility is ready for use. The users guide has been prepared for input processing and reports. Existing data files have been converted and loaded to the database.

2.10.14. P- Operational test is complete. The batch system has been turned over to the users.

2.10.15. Q- Conversion specifications are complete for converting an existing machine-readable file to database input.

2.10.16. R- Definition of batch database reporting functions is complete.

2.10.17. S- Batch reporting functions have been prepared and tested.

2.10.18. T- On-line data entry requirements have been defined.

2.10.19. U- Parameters describing on-line data entry screens have been prepared and stored on PSL.

2.10.20. V- On-line data entry screens have been created and tested. Test data has been stored on PSL.

2.10.21. W- On-line data entry and reporting facility is ready for operational testing. Users guide has been updated to reflect on-line functions.

2.10.22. X- On-line reporting functions have been defined.

2.10.23. Y- On-line reporting functions have been developed and tested.

2.10.24. Z- On-line facility operational testing is complete.

## 2.11. How to Use The Guide.

The Guide is organized in a sequential step approach. Its organization encourages the use of methodical, step-by-step progress from the beginning request by a prospective user to the development of the database subsystem which provides the requested support.

The Guide does not separate those tasks performed by user personnel from those performed by DA personnel. It, instead, identifies who performs each task in the development path. It is the responsibility of user and DA personnel to work closely together, keeping the progress of development flowing smoothly.

It is important, at this time, to point out that the effective and productive database design requires strict adherence to established standards and conventions. These standards were established to make expansion of the database easier and less costly. They also affect the ability of the database to be utilized by other users. When conditions or data are encountered which are not standardized, standards may be defined and recommended for adoption. Refer such situations to the DA staff for action.

Complete and accurate descriptions and comments are essential to the effective definition and understanding of the proposed system. As many comments as are necessary to fully describe the system, its functions, purpose, scope, and data content should be provided. The data dictionary will hold all of the information describing the application. It is, however, not possible for the dictionary to contain and assist in the definition of the system when the user does not enter the necessary data but keeps it in his/her head. The final product will only be as good as its original definition.

Section 3 of the Guide and related appendices provide detailed instructions for each step of the database design and implementation. It is recommended that those sections and appendices which apply to individual members of the

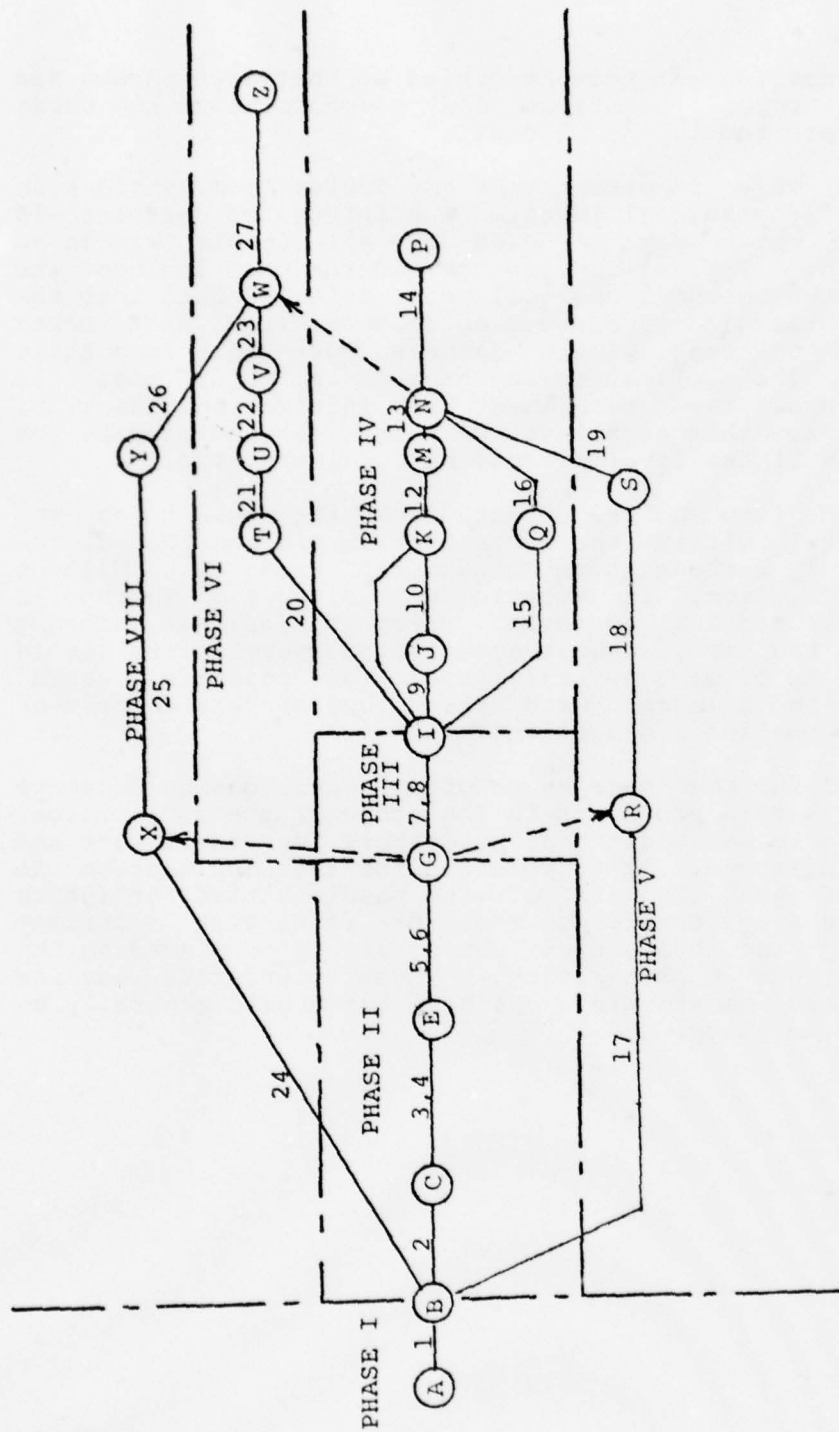


design and development team be copied so that each person has a personal copy. Questions and comments about the Guide should be referred to the DA staff.

It is very important that the design team consider an application from several levels. The integrated database is a resource which may be used by all levels within an organization. Typical design considerations include the needs of the persons who will enter detailed data into the database. What kind of support do they require? What makes the effort of feeding the database worthwhile from their standpoint? The application may have as a major goal the support of upper level management with information. However, it is necessary that each level between those who prepare the data and the higher level of user be considered too.

When the flow and use of data from the lowest to the highest level within the organization is considered, the database truly becomes an organizational resource. Without this consideration, it becomes a limited tool which will never achieve its full potential. Keep in mind that storing data in its raw form, without summarizing, permits its use in a wide variety of ways by all levels of potential users. This must be balanced with the volume of data to prevent overloading on-line storage facilities.

One of the phenomena which often occurs during database design is a sudden expansion in the scope of the application. It is common to see additional potentials for development and use of the database. It is possible for the application to get out of hand as all of the possibilities for future capabilities are identified. Therefore it is very important to identify the boundaries which are to be placed on the application under consideration. These boundaries may be moved to accommodate minor changes, but should generally be firmly held in place.



### 3. DETAIL DESIGN AND DEVELOPMENT INSTRUCTIONS

Each of the summary steps described in Section 2 is described in detail in the sections below. The appendices provide additional clarification of specification and documentation preparation.

Before proceeding, an understanding of the database development philosophy is essential. The methods employed to achieve an operational database are designed to support the basic development philosophy.

Databases are developed to serve users of the information which will be contained in the database. The user's needs must be met in a timely and effective manner. Success or failure of the application may rest primarily on the timeliness of support rather than upon its ultimate efficiency. Few users, unless they are billed for their computer time, are concerned whether a system requires one minute or one hour to process their data. Their only concern is that the end results, the bottom line, are achieved within the time frame in which they need support.

While this philosophy can be carried to the extreme where all users are hurt by inefficient processing approaches, a middle ground is required which gets the user what they need without driving the computer into the ground.

Integrated database design and development is not an overnight task, even for small applications. Careful, methodical analysis is required. Extensive research into the current and potential uses of the database is required to establish a database which is responsive to current needs and, at the same time, flexible enough to avoid becoming a maintenance nightmare as the inevitable changes arrive.

It is difficult to compress the initial steps of database design, the element/group/entity relationship definition process. This is the foundation of the entire development and it must be laid carefully. However, it is possible to dramatically compress the remainder of the development effort through use of the Universal Database approach. UDB represents the ability to implement a new database facility with a minimum of analyst/programmer resources in a very short period of time.



Of primary importance is the ability of the UDB database to be enhanced, augmented, and changed without extensive effort, reducing the long-term maintenance costs significantly.

The UDB provides a number of standard features and facilities to the user. These features are included in every UDB and are supported by all available software. Features and facilities are:

- a. Multiple databases may be established using the same software. Each database is identified by a unique 8-character code.
- b. Each database is logically independent of other databases in the same storage area. Even databases performing identical functions for different users are isolated from each other unless the users desire to link the databases together.
- c. Links between record types of one database to record types of another database can be made if desired.
- d. Each database is fully supported by IP software. Specialized software IP's are generated by a parameter-driven program which creates COBOL IP source programs. See Appendix A.12.
- e. Each record occurrence within a database contains the database identifier to maintain integrity and security control.
- f. Each database user is identified by organization, name, and password. Update and access authorizations are defined for each user. IP software automatically verifies the user's authorization to update the database and rejects unauthorized updates.
- g. Audit totals are maintained on each database. These include:

1. The date the database (and each record occurrence) was created.
  2. The date the database (and each record occurrence) was last updated.
  3. The total number of entry records stored in the database.
  4. The total number of entry updates performed on the database.
  5. The security classification of the database.
- h. Keyword cross-references may be established to any database entry occurrence. Definitions may be supplied to keywords and keywords may be related to each other, e.g., ship to submarine.
  - i. A reference table may be established to any database for look-up requirements. This table contains provision for a directly-accessed identifier, high and low equivalent values, and an expanded description of the table entry.
  - j. New data elements may be added at any time. Where specialized IP software and/or on-line retrieval is used, schema and IP software changes are accomplished within a few hours.
  - k. Data elements may be expanded, contracted, or deleted with minimum effort. Typically, file maintenance changes are made using CULPRIT to create IP entries relocating or altering the database record data elements.

### 3.1. PHASE I. INITIAL REQUIREMENT DEFINITION.

Phase I begins with the receipt of a user request for database support. This request should be received as defined by NAVINTCOM Instruction 5230.4C of 26 September 1978. Ad hoc retrieval requests for database reports are not addressed

by this Guide unless the service will be of a continuing nature.

Upon receipt of the service request, the appropriate project manager will contact the user's point of contact and review the scope of the requested services. A determination will be made whether the service can be supplied by an existing database facility or by modifying an existing facility. If neither of these options is practical, a new database facility will be considered.

Where the scope of the services desired is such that long-term (greater than 6 months) resource allocation will be required, a service analysis will be performed. Service analysis techniques are discussed in Appendix A.10.

The DA staff and the project manager reviews the requested services against the long-term development plans to insure that consistency is maintained. Where the service is not validated by long-term planning, it must be further reviewed by that group prior to proceeding with database development.

The project manager will determine if the user is identified to the data dictionary. Appendix A.3 defines the procedures for adding a new user to the dictionary. Where the user will be supported by an existing database facility, the user's authorization is updated by procedures defined in Appendix A.5.

The development of a new database facility requires that a subsystem description be prepared. Appendix D provides guidelines for the content of the subsystem specification which is stored in the data dictionary using procedures described in Appendix A.4.

The DA staff prepares an estimate of the time and resource requirements for the entire project. The estimate is compared against other projects to identify potential conflicts which could necessitate rescheduling of projects. The user identifies funding and formally approves the project.

### 3.2. PHASE II. DATABASE FOUNDATION DESIGN.

This phase of the database design is concerned with the definition of data elements, element groups, and element group entity relationships. Each step of this phase must be

performed carefully and methodically to insure an effective design.

Of the five steps in Phase II, the user's ADP analysts perform three. The remaining two are performed by the DA staff.

Step 2, the initial data element definition, is performed by user analysts. Appendix A.6 provides the guidelines for this step. During this step there are several things to keep in mind which will improve the resulting definitions:

- a. The user and individual user personnel will be thinking in terms of their primary service request during interviews. While this is most important to them, it must also be a point of departure for the analyst. Often the needs of the user are subtle, surfacing only partially in the requested application. It is necessary to probe the underlying reasons for the requested service.
- b. Users familiar with batch file systems have had ADP serving a single purpose and often a single user. Integrated database is ideal for serving multiple users with the same information IF the database is designed to work that way. The databases developed under the Guide will be designed, wherever possible, to serve multiple users and multiple levels within the same user organization.
- c. Many data elements which the user desires to see in reports are actually the result of interaction between several other elements. When an element is nothing more than a summary of several others, the other elements are the ones to be in the database. The composite element is useful for report outputs but is actually redundant data. There are times when such redundancy is justified but it must be clearly explained when describing the element.



- d. It is important to identify those elements which the user will frequently use to access the database information. These elements will often become key fields in the records and will be handled differently during the final stages of design.
- e. The use of data forms other than display numeric or alphanumeric is not encouraged. These two forms are the only ones which are reasonably compatible across hardware lines and this compatibility will become more important in the future as distributed databases are used. Computational (binary integer) elements may be defined where this is desirable from a processing point of view but must be completely described in the comments section.

While a number of these things may seem obvious, they actually represent a significant number of error conditions or reworking efforts that have occurred during actual database designs.

Step 3, the DA staff review, is performed by NIPSSA personnel. User analyst personnel should be available for consultation when questions arise. The DA staff review consists of:

- a. A review by the project manager. The project manager reviews each element description for:
  - 1. Syntactical correctness. Spelling errors, missing quote marks, and omitted periods are common errors.
  - 2. Redundant elements. It is common for two or more identical elements to be present under different names, e.g., NAME-PERSON and POINT-OF-CONTACT.
  - 3. Incorrect descriptions. Numeric elements must have PICTURE 9 and value zeros, etc.



4. Incomplete definitions. The definition is inadequate to describe the element, its purpose, and the way it is used.
5. Variation from standards. The element is defined in one of the standards documents listed in Appendix E.2 and does not match that definition.
6. Variation from conventions. The element is defined in Appendix B, usually generically, and does not follow those conventions.
7. Security classification conflicts. The element is assigned a security classification inconsistent with its description.

Variation from standards (5 above) is one area where it is difficult for user analysts, particularly contractors, to catch errors. Standards manuals are typically in short supply. It is assumed, therefore, that a number of these type errors will occur. The remaining errors should be caught by the analysts prior to submission for DA review.

b. Review by the database administrator (DBA).  
The DBA reviews the element definitions for:

1. Conflicts with definitions of elements already present in the database.
2. Conflicts with element names already in use within the database.
3. Potential fits within existing database entity relationships.

When the DA review is complete, the project manager prepares a summary identifying the errors or potential problem areas. This summary is presented to the analysts for correction during step 4.

Step 4, group elements, is performed by user analysts following correction of errors and discrepancies found during the DA review.

Grouping of elements is often accomplished easily by simply clipping together the coding sheets for associated elements. Standard groups, such as dates, should be associated first. Appendix A.7 describes the procedure for preparing IDD entries for data element groups. Once the IDD entries have been prepared, they should be loaded to the PSL after the last elementary data element entry.

Step 5, group element review, is performed by the DA staff. The process described for step 3 above is repeated. In addition to the items reviewed during step 3, the project manager checks:

- a. The logical ordering of elements within the groups. Elements are checked to be sure they are organized in the most effective manner for database utilization.
- b. Groups follow established conventions as defined in Appendix B.
- c. Where groups are addressed by existing standards, they conform to those standards.
- d. The aggregate security classification of a group is no less than the highest classification of any individual element within the group.

Once finished with the group review, the project manager refers the element groups to the DBA who:

- a. Reviews each element group for its anticipated location in the database definition. This information is added to the comments associated with the element.
- b. Determines if any conflicts remain/exist between the proposed element groups and the existing database.

The project manager prepares a summary of discrepancies as in step 3 and reviews the discrepancies with the user analysts.

Step 6, data entity relationships, is performed by the user analysts. This step is the last one in the definition process. The elements and element groups defined previously are associated into logical data entities. These entities will appear as record types to the end user.

Previously flagged elements which have been defined as residing in the database are handled first. These elements/groups are separated from the remaining definitions. Logical associations which existed to these elements are reviewed. This review:

- a. Determines whether an associated element is so closely associated that it should be located in the existing database entity rather than in one specifically defined for the user's application. For example, the stock exchange designator for a company is as basic an identifier of the company as its name. Therefore it would be logical to add that designator to the database entity in which the company name appears. The number of shares outstanding, on the other hand, is more loosely associated and should be located with other share-oriented information.
- b. Identify the elements/groups which will form the access key to a new database entity. A maximum of 32 characters may be used for this purpose. Organize the elements/groups in the order of their importance so that they will appear in descending order from left to right. Create a group definition for these elements/groups with a prefix of "ID-".
- c. Assign a name to each database entity defined. The name should be user-understandable.
- d. Define a maximum of two elements/groups within the entity which may be accessed on a direct basis. These two fields, if used, will be the subjects of secondary indexes.
- e. Order the remaining elements and groups within an entity in the logical order described previously.

- f. When elements or groups will repeat an unknown number of times, these elements/groups MUST be defined as separate database entities. Frequently such a definition has no logical key fields which may be used to uniquely identify each occurrence. When this occurs, the analyst may have to create a pseudo key. For example, a date and a sequence may be used to define comments information associated with another entity.
- g. Elements or groups which repeat a known number of times may be associated with other elements/groups in an entity. The repeating group will be placed at the logical end of the entity.

CAUTION: Be sure that the repeating groups do not contain data values which will be repeated in other occurrences of the entity. Such a condition introduces redundant data into the database. When this condition is present, effective scanning of the database for those values in the embedded repeating group is not possible. Such a situation should be avoided except under VERY unusual conditions. Refer such situations to the project manager.

- g. When a database entity has been defined, it is possible that it may consist of three parts:
  - 1. The fixed data. This group is present at the front of every record type in the database. It must be defined as a group element with a prefix of "FI-". The fixed group always consists of the same six data elements in the same order. These are: (a) CLASS-xxxx; (b) HANDL-xxxx; (c) DATE-MOD-xxxx; (d) DBID-xxxx; (e) LOG-TYPE-xxxx; (f) DATE-STORED-xxxx (xxxx is the IDMS schema identifier number of the record type).



2. The identifier. This will always occur. It must be defined as a group element with a prefix of "ID-".
3. The body of the database entity. This will always occur. All remaining elements/groups assigned to the entity are placed in this section. It must be defined as a group element with a prefix of "DA-". This section has a total length of 800 bytes. A filler is added to the end of the section to achieve a total of 800 bytes.

Appendix H illustrates the element definition process.

### 3.3. PHASE III. DATABASE INTEGRATION

During Phase III, the definitions of Phase II are integrated into the database structure. Two task steps accomplish the integration. Both are performed by the DA staff.

- a. Step 7 performs a final review of the Phase II results:
  1. Element entities are assigned to locations within the database structure. Elements to be merged with existing database entities are noted.
  2. The associations effort performed in Step 6 is validated and final modifications made as required.
  3. The PSL is reviewed for the last time and corrections made where necessary.
  4. The PSL is transferred to the data dictionary.



5. Data dictionary reports are prepared and reviewed to insure that the new application does not conflict with the existing database definition. Corrections are made as required.

b. Step 8 updates the UDB schema:

1. The schema entry in the PSL is updated, adding copy statements to include the new application.
2. The schema, DMCL, and subschema is compiled using the standard JCL defined in the DBA guide.
3. The subschema definition is reproduced for use by user analysts during subsequent phases of the development.

#### 3.4. PHASE IV. BATCH INPUT PROCESSING AND FILE CONVERSION DEVELOPMENT.

This phase of the application project begins actual development of software to support the defined database. Software developed during Phase IV will be used to load data to the database through batch processing. It is assumed that data will be presented to the system in card-image form whether prepared on punched cards, magnetic tape, or disk.

Where an existing automated or machine-readable source of information will be converted and loaded to the database, it is necessary that conversion software be prepared in addition to writing batch IP's. Most conversion requirements can be effectively met through use of CULPRIT as the conversion processor.

The functions of this phase are performed by user analysts. Regular consultation with the project manager is encouraged. Prior to beginning the phase, the project manager:

1. Assigns a block of IP identifiers (defined by the DBA) to the application following

established conventions defined in Appendix B.6.

2. Defines the PSL which will be used for IP parameter definitions and, if required, conversion software.
3. Defines the PSL which will contain test data for the application.
4. Assigns personal identifiers to each user analyst who will require computer time.

Step 9 of the application development defines the input processing formats. These basic constraints are applied to each IP format designed:

1. Each IP format is processed as a separate transaction, independent of the transactions preceding and following.
2. Each data element processed by an IP must be contained entirely within the limits of the IP medium.
3. Each IP reserves the first four positions of its line as an identifier.
4. Every IP identifier must be unique.
5. Every IP must establish currency with the database record occurrence upon which it will act.
6. With few exceptions, an individual IP acts directly against only one database record occurrence. Each exception must be approved by the project manager and documented as part of the IP comments in the IDD.

These conventions apply during the definition of the IP format:

1. Data required to establish currency with the database record occurrence to be processed are placed immediately following the format identifier.

2. If data will be coded from a printed source document, the element sequence will be established to follow a left to right, top to bottom scan of the source document where possible.
3. When the IP coding sheet will be the source document, the most frequently used fields should precede less frequently used fields.
4. Blank columns may be left adjacent to fields where future expansion of the field is anticipated. However, it is better to include the expansion into the definition initially to prevent recompiling the IP at a later date. The coding form may omit the unused positions and the IP instructed to zero or space fill the data.
5. IP identifiers must be assigned in a specific sequence. The processing module presorts incoming data into IP identifier sequence to improve processing efficiency. Therefore, it is essential that the IP which initially establishes a database record occurrence alphabetically precede another IP which modifies the same record occurrence. For example, the AATS IP stores a NIM-PROJECT record while AAUM and AAWM IP's modify the NIM-PROJECT record.

Once the physical layout of the IP has been defined, the layout is described in IDD form as described in Appendix A.12. These descriptions are prepared in card image form and delivered to the project manager for loading into the data dictionary. The project manager:

1. Checks the description for correct syntax and organization.
2. Insures that all data elements/groups used by the IP are present in the data dictionary and belong to the application.
3. Delivers the description to the data dictionary librarian for loading.

4. Informs the user analysts when dictionary loading has been completed and provides a dictionary report of the IP descriptions.

The data dictionary librarian:

1. Enters the descriptions into the data dictionary.
2. Corrects minor errors. Refers other errors to the project manager.
3. Prepares a report containing the descriptions loaded.

Step 10 is the preparation of IP parameter statements for each defined IP. This step is performed by user analysts. This step may be performed in parallel with step 9 above.

IP parameter preparation is also described in Appendix A.12. Appendix J.1 describes the JCL required to store parameters in a PSL. Once stored on PSL, IP's may be compiled using JCL described in Appendix J.2.

Step 11 is the most time-consuming step of this Phase. Comprehensive test data must be prepared for each IP. Test data must:

1. Test each IP for all possible DATABASE conditions which can affect IP accuracy:
  - a. Presence of a record occurrence in the database when attempting to store another occurrence with the same identifying data.
  - b. Attempt to modify or delete a record occurrence which is not in the database.
  - c. Attempt to associate one database record occurrence with another when one or both of the occurrences are not present in the database.
  - d. The database is full.

All of these conditions are reflected by error codes returned by IDMS following a DML command. The standard logic of an IP will catch most of the errors. However, it is necessary that tests be performed to insure that this type of error condition is caught.

2. Test each IP for all possible DATA CONTENT conditions which can affect IP accuracy:

- a. Each individual data field must be tested for both good and bad data. Where a field is range-tested, tests for values preceding and following the valid range as well as tests of the edges of the valid range must be made.
- b. Combinations of data elements which are interrelated, such as dates, must be tested with combinations of good and bad data so that all possible combinations are checked.
- c. Each field must be tested for asterisk delete. Those which do not allow asterisk delete must produce the proper error message.
- d. Each field which permits the use of a question mark to optionally omit data must be tested with both the question mark and blanks (an error condition).
- e. Each numeric field will be tested by placing alphabetic and blank characters in each of the fields positions in individual tests.

Prepared test data will be stored on the PSL designated by the project manager, listed and provided to the project manager for review.

The project manager:

1. Reviews the test data to be sure it follows basic IP conventions.



2. Establishes a test database for use in testing IP's.
3. Defines those analyst personnel who will be authorized to use the test database to the UDB system.
4. Defines the IP's to the system so testing can be performed.

Each IP is thoroughly tested in step 12. All test data is applied and error conditions reviewed. Processing errors are corrected and tests rerun. Test results are reviewed by the project manager.

The initial subsystem users guide is prepared in step 13. Sections 1 and 2 of the users guide are prepared similar to those shown in Appendix F. Individual coding sheets and instructions are prepared for each IP generated. Additional coding sheets and instructions are prepared for each association (connect and disconnect) function required using standard association IP's.

Batch input coding instructions will be grouped for logical use. For example, those instructions for the initial storing of a database record occurrence are grouped together as are those to modify the database. Frequently used coding instructions are placed in a group at the beginning of the section.

Separator tabs aligned to groups of functions will be provided to make locating specific coding instructions easier. The page location of each instruction will be provided in the table of contents.

Step 14 is the operational test of the batch system capability. The user personnel who will utilize the system are trained and assisted in initial use of the system. A training class of one day duration (maximum) will be held for each of these user areas:

1. User management. This class will describe the system capabilities from a management viewpoint. The part the system will play in improved management of security will be emphasized. Visual aids and handouts will be prepared to acquaint management with the

system and its interface with other facets of NICOLS.

2. User personnel who will operate the system. This class will describe the operation of the system in more detail. The users guide will be thoroughly reviewed. These points will be covered:
  - a. JCL and job submission.
  - b. Data entry preparation.
  - c. Coding conventions where they exist.
  - d. Report preparation.

Visuals aids and handouts will be prepared to support the training.

Step 15 defines conversion criteria for each of the existing machine-readable files currently in use. This information includes:

1. The source file and location of each data element.
2. The object IP format and the position of each data element.
3. Differences in length and mode of each data element, where occurring, between source file and database.
4. Conversion requirements, where required, of each data element. This includes translation of codes, special verification of source data, and optional purging of data.

The conversion software is prepared in step 16. Using the conversion criteria prepared in the previous step, CULPRIT modules are prepared to effect the conversion of existing machine-readable files into IP format.

The files are converted and the output listed prior to loading of data. Listings are reviewed by insure

conversion criteria have been met. Upon approval of the project manager, the converted files are loaded to the operational database.

### 3.5. PHASE V. BATCH OUTPUT SUPPORT DEFINITION AND DEVELOPMENT

This phase of the application development is accomplished in three steps:

- a. Step 17 expands on the service analysis performed during step 1 and defines the output services in detail. This includes identifying the format of outputs and the database resources required to satisfy the output service.
- b. Step 18 converts the specifications of step 17 into programmed output capabilities. CULPRIT and, where required, COBOL report writer, are used to produce and test each output service.
- c. Step 19 provides updates to the users guide defining each output service, illustrating the output, and describing the procedures for obtaining the output.

Step 17 expands upon the service analysis information placed in the data dictionary. The physical layout of the report is prepared. These factors are considered:

- a. Security classification of the output.
- b. Distribution of the output and the knowledge of its anticipated audience.
- c. The medium upon which the output will be presented.
- d. The use of cover pages on printed reports.
- e. The conversion of encoded data elements within the database to expanded and more readable descriptions within the reports.

- f. Frequency and sensitivity of the output preparation.
- g. Computations performed as part of the output.
- h. Data restrictions (need to know).

A comprehensive narrative description of the output will be added to the service analysis description and stored in the data dictionary. Complete pictorial presentations of printed or displayed reports, cover pages, and associated displays will be prepared in sufficient detail that development of the capabilities in the following step will require little, if any, additional analysis.

Step 18 uses the comprehensive analysis and definition of the output capabilities defined in step 17 to produce software necessary to satisfy the desired output service. CULPRIT will be used wherever possible for preparing output facilities to be executed in a batch environment. COBOL Report Writer may be used with the approval of the project manager.

Each output capability created will be thoroughly tested. Test results will be reviewed by the project manager and compared against the specifications of the requested service.

Step 19 provides the user with instructions and examples of the utilization of the output service. Each output service will be individually documented and placed in the users guide. Each entry will contain:

- a. Instructions for requesting the output service, including a description of the JCL and control cards required. Control card fields will be described in detail. Illustrations of typical control card configurations will be included.
- b. Definitions of options associated with the output service, if any. Where options are available, each will be described separately and illustrated.
- c. An illustration of the normal output of the

service where the output is visually presented.

At the completion of this phase, a complete batch processing capability of the application is available to the user.

### 3.6. PHASE VI. ON-LINE INPUT PROCESSING.

On-line input processing provides application users with the ability to perform data entry and correction from an interactive CRT. This phase is similar to Phase IV except that data conversion and initial users guide preparation have already been performed.

Phase VI is composed of five steps:

- a. Step 20 defines the format of on-line input processing screens after having determined what data entry and corrections requirements are present for the application. Not all application data may be suited for on-line input processing.
- b. Step 21 prepares on-line input processing program parameters based on the specifications developed in step 20. The parameters are reviewed and stored on PSL in preparation for compilation.
- c. Step 22 utilizes the parameters developed in the previous step and creates on-line input processing programs for data entry. Each program is thoroughly tested using test data developed in this step. Corrections are made as required and the PSL updated. Test data descriptions are stored on a designated PSL for future use. The project manager reviews test results against the specifications to insure correct processing.
- d. Step 23 updates the users guide to include an instruction for each on-line input processing screen. Instructions will identify each data element used, all processing options, function key usage, and illustrate each



screen.

5. Step 27 is normally performed after completion of Phase VII (on-line output services). This step includes the training of user operating personnel, development of training materials, and direct user assistance for a short period of time.

Step 20 is the on-line input processing definition step. Service analysis information is reviewed and requirements for on-line data entry and correction/modification services are translated into specifications for teleprocessing screens. The physical layout for the screens is prepared and reviewed. Descriptions of screen functions are prepared and added to the service analysis description. The logical progression from one screen to another where screen functions are related is defined and function keys assigned as required.

Step 21 utilizes the descriptions prepared in the previous step to generate on-line IP parameters and data dictionary entries. Appendix A.14 provides detailed instructions for preparation of screen parameters.

Step 22 compiles the screens from the parameters stored on PSL and in the data dictionary. Each screen is tested using a set of test data developed for the purpose. Test data will be developed using the same criteria as in step 11 (batch test data preparation). Test data descriptions will be stored on PSL in narrative form for future use. Test results will be reviewed by the project manager to insure that all requirements have been satisfied.

Step 23 is the preparation of user guide information for on-line input processing. An entry for each screen will be included which:

- a. Illustrates the screen.
- b. Describes each data field to be entered on the screen.
- c. Defines the association of other screens which are logically appended to a screen. This association description will identify the function keys or command action required

to reach associated screens. A pictorial diagram of screen relationship will be included.

Step 27 is the final step of on-line system implementation. Phase VII must be complete, or omitted, prior to performing this step. The step is the operational test of software and training of user personnel. A one-day (maximum) training class for user operating personnel is held to acquaint users with the capabilities of the system. Training materials and handouts are prepared to support the class and are turned over to the project manager for use in future training. The operation of the system is observed and discrepancies noted. Corrections are made where necessary and the system documentation and programs updated. The project manager reviews the system operation and approves the system for turnover.

### 3.7. PHASE VII. ON-LINE OUTPUT DEFINITION AND DEVELOPMENT.

This phase is nearly identical to Phase V except that on-line display output is being developed. On-line display outputs may be developed in a number of different ways:

- a. CULPRIT or COBOL programs may be written which are intended for display use (80-character lines, 20 lines per page) and executed using the batch report capture facility. This facility places captured reports in a disk queue for remote display. This approach is useful when preparing reports which have periodic use or require more computer time than the operator is willing to wait for in interactive mode.
- b. Outputs produced as a result of stored queries processed through ON-LINE QUERY. These may be standing or modifiable queries which are run on an ad hoc basis and require little time to complete.
- c. Directly generated queries using ON-LINE QUERY.
- d. Specially developed COBOL report programs producing specialized reports. These are

typically complex outputs which require decision or arithmetic functions not available in ON-LINE QUERY.

Phase VII steps are:

- a. Step 24 expands on the service analysis definitions of on-line output services. Each requested service is carefully analyzed to determine which of the four development modes described above will be used to implement the service. Detailed descriptions of the services are added to the service analysis prepared in Phase I and the data dictionary updated.
- b. Step 25 is the preparation of output programs or queries to satisfy the service request. Each program or query is stored on a designated PSL. Each capability is tested and test results reviewed by the project manager.
- c. Step 26 is the preparation of updates to the users guide reflecting each of the on-line services provided.

Step 24 requires a thorough review of those service requests prepared during Phase I. Each request which requires on-line output support is evaluated. One of the development approaches defined above is selected for each service requested. The detail definition of the visual display to be produced is prepared. Specifications are expanded to include complete details about the service, its purpose, and the depth of logic required to produce the capability. The data dictionary is updated with the additional information. CAUTION: When updating remarks of the service analysis, first update the information on the PSL and then update the data dictionary from the PSL.

The actual software or query modules are prepared and tested during Step 25. Each output service is thoroughly tested and the results approved by the project manager.

Step 26 is the updating of the users guide to reflect the instructions for use of each service request. The users

guide will contain, for each output service:

- a. A pictorial of the output service.
- b. Complete step-by-step instructions for requesting the output from a CRT terminal.
- c. A description of any options available with the output service. Where the option changes the format or function of the service, an illustration and explanation of each option will be included.

## APPENDIX A.1

### GENERAL INSTRUCTIONS

The methodology of the Design Guide utilizes the Integrated Data Dictionary (IDD) as the repository of all design information. The formats provided with the Guide are tailored from IDD statements to achieve the desired end results. For additional IDD capabilities, the user is referred to: Integrated Data Dictionary User Guide, Cullinane Corporation, Wellesley, Mass.

IDD entries must be prepared on punched cards, or punched card images on standard OS unlabeled magnetic tape, for entry into a program support library (PSL) and from there into the data dictionary. Input should be prepared in the order of the steps within the Guide. This will insure that the proper sequence of information is provided during dictionary update.

A library support feature (PSL) is provided to make storing and modifying the IDD entries easier. This feature permits storing IDD data on a temporary file which can then be readily modified until the IDD entries for the application have been completed and are ready for IDD update. Use of this library facility is described in detail in Appendix A.2.

### General IDD Coding Conventions

1. Certain letters and numbers can be misinterpreted easily. Conventions have been established to reduce confusion to a minimum. The following should be used when entering information on IDD coding forms to minimize possibility of error:
  - a. Numeric zeros are slashed; alphabetic letter "O" is written normally.
  - b. The letter "Z" is slashed across its diagonal bar; the number "2" is written normally.
  - c. The number "1" is underlined; the letter "I" is written with both top and bottom bar.



- d. A single quote mark (') is used for all IDD entries where quotes are required; a double quote mark (") is not used.

2. Comments:

- a. All comment lines begin in column 12 of the punched card and require a single quote mark in column 12.
  - b. Comments which require more than one line may be continued indefinitely. Each continuation line must have a hyphen (-) in column 7.
  - c. The last comment line must be terminated by a single quote mark.
  - d. If a comment includes apostrophies, they are seen by the IDD as a single quote mark. To prevent termination of the comment, two single quote marks (apostrophies) are coded together in successive card columns (''). For example: "The user's guide is....." is coded as: 'THE USER'S GUIDE IS....'
  - e. ELEMENT DESCRIPTION, RECORD DESCRIPTION, and ELEMENT DEFINITION clauses are coded as if they were comments. The ELEMENT DESCRIPTION and RECORD DESCRIPTION clauses are limited to 40 characters.
3. A standard program coding form, such as the COBOL form, may be used to prepare IDD entries. IDD statements must not begin before column 7 or extend beyond column 72.
4. It is recommended that at least two spaces separate clauses on the same line and that no more than two clauses be placed on the same line. Readability of the coded data is

improved if the sequence of the statements is not otherwise changed. Appendix G contains the IDD syntax requirements for reference purposes.

5. A period (.) must terminate all IDD entries. Periods must not be placed at the end of individual clauses. Each coding form indicates where the period must go if the entry is completely coded. If the entry is shortened, the period must immediately follow the last clause in the entry.
6. The SAME AS ELEMENT clause is a powerful time-saving tool. This permits copying the description of another data element which is basically the same as the element being defined (such as month), eliminating repetitious coding. Any clause which is different from the element being copied, such as the description (long name) or definition, is coded. The resulting element definition is consistent in format to similar elements while identifying those unique features of each individual element. Appendix E.11 illustrates predefined data elements which are frequently copied.

## APPENDIX A.2

### PROGRAM SUPPORT LIBRARY (PSL) USE

The IBM 360 Operating System provides a utility function which permits the assignment of specific temporary library areas. IDD input data may be stored in this library and then massaged until it is ready for updating into the data dictionary.

As punched card images containing IDD update information are prepared by the user, the project manager will assign a PSL location for storage of the information. The user is provided with a processing job control deck to permit updating of the library area (member).

THE USER MUST EXERCISE EXTREME CAUTION IN PERFORMING UPDATES OF THE LIBRARY MEMBER TO AVOID DESTROYING THE DATA. CARDS USED IN EACH UPDATE SHOULD BE RETAINED UNTIL SUCH TIME AS THE PSL IS TRANSFERRED TO THE DATA DICTIONARY.

#### Composition of the User's PSL Job Control Deck

Figure A.2.1. illustrates a typical PSL job control deck. The cards in this portion of the deck should not be modified in any way by the user. Columns 73 through 80 of the JCL deck are punched with an identifier and sequence numbers to assist the use in keeping the deck in proper order.

```
//B78xxx01 JOB (Hxxx,0000,5,5,,,,,60),'PSL FOR DB DESIGN',  
//      MSGLEVEL=(0,0),CLASS=E,REGION=65K  
//STEP      EXEC PSLUPDTA,MEMBR=applic  
//UPDT.SYSIN DD      *  
./      CHANGE NAME=xxxxxxxx,LEVEL=00,SOURCE=0,LIST=ALL  
./      NUMBER NEW1=10,INCR=10  
.  
.  
.  
update data  
.  
.  
./ ENDUP  
/*
```

Figure A.2.1.

"applic" is the name of the PSL library being processed. The library name is assigned by the DA staff.

"xxxxxxx" is the name of the used-defined PSL entry being processed. "B78xxx01" is the name of the job, where xxx is replaced by the assigned person identifier.

"(Hxxx,0000....)" is accounting information, where Hxxx is replaced by the defined location code and person identifier, and 0000 is replaced by a room number or other building locator code.

#### Modifying The Contents of The PSL Member

Each statement stored in the PSL member is assigned a sequence number. This number is placed in columns 73 through 80 of the statement by the utility program when the data is originally stored in the PSL.

Any statement may be modified by punching a replacement statement with the sequence number of the PSL member statement in columns 73-80. Figure A.2.2 illustrates a PSL member. Figure A.2.3 illustrates modification statements which are to be applied against the PSL member. Figure A.2.4. illustrates the result of the modification after updating the PSL member.

ADD ELEMENT NAME IS SEC-HANDLING	00000010
PREPARED BY 'LET'	00000060
ELEMENT DESCRIPTION IS	00000110
'RELEASABILITY CODE'	00000160
PICTURE IS XX USAGE IS DIAPLAY	00000210
VALUE IS SPACES	00000260
ELEMENT DEFINITION IS	00000310
'THE FIERD DEFINES STANDARD DOD	00000360
- 'RELEASABILITY CODES'	00000410
USER IS 'NIPSSA51 TOWNER'	00000460
RESPONSIBLE FOR DEFINITION.	00000510

Figure A.2.2.

PICTURE IS XX USAGE IS DISPLAY	00000210
'THE FIELD DEFINES STANDARD DOD	00000360
RESPONSIBLE FOR DEFINITION.	00000510

Figure A.2.3.

ADD ELEMENT NAME IS SEC-HANDLING	00000010
PREPARED BY 'LET'	00000060
ELEMENT DESCRIPTION IS	00000110
'RELEASABILITY CODE'	00000160
PICTURE IS XX USAGE IS DISPLAY	00000210
VALUE IS SPACES	00000260
ELEMENT DEFINITION IS	00000310
'THE FIELD DEFINES THE STANDARD DOD	00000360
- 'RELEASABILITY CODES'	00000410
USER IS 'NIPSSA51 TOWNER'	00000460
RESPONSIBLE FOR DEFINITION'.	00000510

Figure A.2.4.

#### Add New Statements To The PSL Member

The user may add new statement to the PSL member by punching the new statements and an identifier sequence number which falls between the two PSL statements where the new statement belongs. Figure A.2.5 illustrates statements to be added to the member shown in Figure A.2.4. A "NUMBER" card (See Figure A.2.1) renumbers the statements. Figure A.2.6 shows the results of the update.

ENTRY-SECURITY IS ENTRY-UNCLASSIFIED	00000270
DATA-SECURITY IS DATA-UNCLASSIFIED	00000271

Figure A.2.5.



ADD ELEMENT IS SEC-HANDLING	00000010
PREPARED BY 'LET'	00000060
ELEMENT DESCRIPTION IS	00000110
'RELEASABILITY CODE'	00000160
PICTURE IS XX USAGE IS DISPLAY	00000210
VALUE IS SPACES	00000260
ENTRY-SECURITY IS ENTRY-UNCLASSIFIED	00000310
DATA-SECURITY IS DATA-UNCLASSIFIED	00000360
ELEMENT DEFINITION IS	00000410
'THE FIELD DEFINES STANDARD DOD	00000460
- 'RELEASABILITY CODES'	00000510
USER IS 'NIPSSA51 TOWNER'	00000560
RESPONSIBLE FOR DEFINITION.	00000610

Figure A.2.6.

#### Removing Statements From the PSL Member

The user may remove statements from the PSL member by preparing a DELETE statement and identifying the statement(s) to be removed. Figure A.2.7 illustrates DELETE statements to delete one statement and three statements from Figure A.2.6. Figure A.2.8 shows the result of the update performed.

```
./ DELETE SEQ1=60,SEQ2=60
./ DELETE SEQ1=210,SEQ2=310
```

Figure A.2.7.

ADD ELEMENT NAME IS SEC-HANDLING	00000010
ELEMENT DESCRIPTION IS	00000060
'RELEASABILITY CODE'	00000110
DATA-SECURITY IS DATA-UNCLASSIFIED	00000160
ELEMENT DEFINITION IS	00000210
'THE FIELD DEFINES STANDARD DOD	00000260
- 'RELEASABILITY CODES'	00000310
USER IS 'NIPSSA51 TOWNER'	00000360
RESPONSIBLE FOR DEFINITION'.	00000410

Figure A.2.8.

### Organizing The PSL Update Statements

The PSL member containing the IDD source data is organized in sequential order. It is necessary that modification statements also be organized in sequential order, ascending from one upwards. The utility program which updates the PSL will prevent completion of any update which is out of order.

Plan updates of the PSL member so that no more than 250 update statements are processed at a time. This will speed up the update process and reduce the chance of errors caused by incorrect sequential organization of the update data.

## APPENDIX A.3

### NEW IDD USER ENTRIES

Each user of the database is an unique entity. A user for this purpose is defined as an organizational entity, of varying size and structure, which is identified individually to the system. Each user may include one or more persons who are identified to the data dictionary as subsets of the user organization. Access to the database is limited to those user personnel who are identified to the dictionary.

Each user is assigned a 16-character abbreviation of the full organizational name. The name of the person(s) within the user organization are each assigned a 16-character name field in the entry. One entry is prepared for each organization/person identified. Figure A.3.1 is an example of the user entry.

1. ADD USER NAME IS. [REQUIRED] The user name is composed of two segments, each 16 characters in length. The entire 32 character name is enclosed in single quote marks. Segment one contains the alphanumeric acronym/abbreviation of the user organization name. The second segment contains the last name of the individual person, left justified, within the user organization. The complete user name must be unique. The name will be assigned and verified by the DA staff. Access to the database is restricted to user personnel identified to the data dictionary. If the person name segment is omitted, the organization as a whole is identified. This serves to define an organization to the data dictionary but does not grant database access.

(8)

ADD USER NAME IS 'NIPSSA51 TOWNER'

2. OF SUBSYSTEM xxxxxxxx VERSION nnnn. [REQUIRED] This clause is completed only if the new user will utilize an existing application system. The DA staff will enter the correct version of the subsystem. This line is omitted if the application is new.

(12)

OF SUBSYSTEM NIMIS VERSION 0101

3. USER DESCRIPTION IS. [REQUIRED] This entry permits expanding the name of the user organization. A single quote mark is placed at the end of the expanded name.

(12)

USER DESCRIPTION IS  
'HEAD, SYSTEMS MANAGEMENT DIVISION'

4. ENTRY-SECURITY IS ENTRY-. [REQUIRED] This statement identifies the security classification associated with the information about the user organization recorded in the data dictionary. Valid entry values are described in Appendix E.1.

(12)

ENTRY-SECURITY IS ENTRY-UNCLASSIFIED

5. DATA-SECURITY IS DATA-. [REQUIRED] This statement identifies the maximum security level of data which the user is authorized to access. Valid entry values are described in Appendix E.1.

(12)

DATA-SECURITY IS DATA-UNCLASSIFIED

6. COMMENTS. [REQUIRED] This statement permits a description of the user organization and location. An unlimited number of lines may be coded as long as they are all contiguously stored in the dictionary. A single quote mark and a period must terminate the last line of comments.

(12)

COMMENTS  
- 'ROOM 244, HOFFMAN BUILDING 1  
- '2461 EISENHOWER AVE.  
- 'ALEXANDRIA, VA 22331'.

ADD USER NAME IS 'NIPSSA51 TOWNER'  
OF SUBSYSTEM NIMIS VERSION 0101  
USER DESCRIPTION IS  
'HEAD, SYSTEMS MANAGEMENT DIVISION'  
ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
DATA-SECURITY IS DATA-UNCLASSIFIED  
COMMENTS  
'ROOM 244, HOFFMAN BUILDING 1  
- '2461 EISENHOWER AVE.  
- 'ALEXANDRIA, VA 22331'.

Figure A3.1.



#### APPENDIX A.4

##### IDD SUBSYSTEM DEFINITION ENTRY

Each major application area of the database is defined as a subsystem. This approach permits separating applications, users, and data into easily managed and controlled entities. Subsystems developed as part of the integrated database are all considered to be a part of the integrated database system as far as the data dictionary is concerned.

Figure A.4.1. illustrates the subsystem definition entries.

##### SUBSYSTEM DEFINITION ENTRIES

1. ADD SUBSYSTEM NAME IS. [REQUIRED] This statement defines the internal name of the subsystem. The name is limited to eight characters in length and the first character must be alphabetic. Blanks and special characters within the name are not allowed. The subsystem name is assigned by the DA staff.

(8)

ADD SUBSYSTEM NAME IS NIMIS

2. SUBSYSTEM DESCRIPTION IS. [REQUIRED] This literal field, enclosed in single quote marks, provides an expanded name of the subsystem.

(12)

SUBSYSTEM DESCRIPTION IS  
'NAVINTCOM MANAGEMENT INFORMATION SYSTEM'

3. ENTRY-SECURITY IS ENTRY-. [REQUIRED] This statement defines the security classification of the subsystem

description entries. See Appendix E.1 for a list of valid classifications.

(12)

ENTRY-SECURITY IS ENTRY-UNCLASSIFIED

4. DATA-SECURITY is DATA-. [REQUIRED] This statement defines the maximum security classification of the subsystem data entries. See Appendix E.1 for a list of valid classifications.

(12)

DATA-SECURITY IS DATA-UNCLASSIFIED

4. COMMENTS. [REQUIRED] This section permits an unlimited free-form description of the subsystem. Its purposes and scope are described in detail. Continuation comment sheets should be used as required as long as all lines are entered into the data dictionary at the same time. The last line of comments must be terminated with a single quote mark. Sections 1, 2, and 3 of the subsystem specifications described in Appendix D are used for guidelines on the information to be entered. These comments are stored in the PSL as a separate member from other IDD data. This permits individual updating of the subsystem specification body. The project manager will provide the appropriate PSL identifier.

(12)

COMMENTS

'THE NAVINTCOM MANAGEMENT INFORMATION SYSTEM (NIMIS)

- 'IS . . . . . '

5. PERIOD (.). A period must terminate the subsystem entry. This period may be placed immediately following the single quote mark at the end of the last comment line.

ADD SUBSYSTEM NAME IS NIMIS  
SUBSYSTEM DESCRIPTION IS  
'NAVINTCOM MANAGEMENT INFORMATION SYSTEM'  
ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
DATA-SECURITY IS DATA-UNCLASSIFIED  
COMMENTS  
'THE NAVINTCOM MANAGEMENT INFORMATION SYSTEM (NIMIS)  
- 'IS . . . . .'

Figure A.4.1.

## APPENDIX A.5

### IDD USER/SUBSYSTEM ASSOCIATION ENTRY

When an existing user of ADP resources requests to use an existing database capability, it is necessary to authorize the user, through the data dictionary, to utilize the facility. Figure A.5.1 illustrates the IDD entries required.

A separate entry must be prepared for the organization and each person within the organization who will be authorized access to the database.

#### USER/SUBSYSTEM ASSOCIATION

1. MODIFY USER NAME IS. [REQUIRED] The user name as defined in the data dictionary is entered.

(8)  
MODIFY USER NAME IS 'NIPSSA51 TOWNER'

2. INCLUDE OF SUBSYSTEM. [REQUIRED] The name of the subsystem supporting the requested facility is entered.

(12)  
INCLUDE OF SUBSYSTEM NIMIS

3. VERSION nnnn. The version number of the current subsystem is entered.

(12)  
VERSION 0101.

MODIFY USER NAME IS 'NIPSSA51  
INCLUDE OF SUBSYSTEM NIMIS  
VERSION 0101.

TOWNER'

Figure A.5.1.



## APPENDIX A.6

### IDD DATA ELEMENT DEFINITION ENTRY

The basis for all ADP system development is the data which will be stored in the database to support the user's requirements. It is essential that each data element within the system be clearly and completely defined. The entries described below provide the facility to enter complete data element definitions into the data dictionary for use in analysis and design of the supporting subsystem.

This step is the most important of the steps resulting in an operational application. It is best performed by the user of the proposed capability who can provide the most detailed information about data elements and their use. For this reason, it is assumed that user personnel not familiar with ADP terms and technology will assist in preparing data element definitions.

Coding sheets, upon completion, should be punched and the statements stored in the PSL using procedures defined in Appendix A.2.

Prior to defining any data elements, the user should keep in mind that the names assigned to the element should be as descriptive as possible, within the limits imposed by the rules and conventions defined in Appendix B. This appendix should be thoroughly reviewed prior to beginning the data element definition process. As the definition and review process proceeds, it may be necessary to change some element names to correspond with existing elements in the data dictionary.

Two textual areas are provided in the definition entries. They are intended for similar but specific purposes. The element definition will be used as the basis for user documentation and coding instructions. The definition shown in this section must be very clear and specific and AIMED AT THE NON-ADP USER at the experience level of the normal data preparer. Comment/definition continuation sheets should be used as required to provide a complete definition of the data element. The comments section contains more specific information about the data

element which is useful to the designer and analyst. Specific ADP terminology may be used in this section of the description. Examples of comments which may be placed in this section are: identification of the element as a major sort field for a report; a key field within the database; a field which requires special table lookups; or a field which requires special processing.

During review of the data elements by the DA staff, additional comments may be entered. Comment/definition continuation sheets should be used as required to provide a complete description of the element.

#### ADD DATA ELEMENT TO DATA DICTIONARY

Figure A.6.1 illustrates the complete element definition.

1. ADD ELEMENT NAME IS. [REQUIRED] This name will be used for all future references to the data element. It is useful to enter the full data element descriptive name (item 4 below) prior to establishing the element ADP name. Using the guidelines provided in Appendix B, define the name within the 16 character limitation. No spaces or special characters may be used in the name.

(8)

ADD ELEMENT NAME IS ORGANIZ-ACRONYM

2. PREPARED BY. [REQUIRED] The initials of the preparer are entered, left justified, in the space provided.

(12)

PREPARED BY LET

3. SAME AS ELEMENT. This field will be entered if it is determined that the element is identical in structure to an element already defined to the database. See Appendix A.1 for a more detailed description of this clause.

(12)

SAME AS ELEMENT ORGAN-ACRONYM

4. ELEMENT DESCRIPTION IS. [REQUIRED] This field contains the expanded name of the data element. More than one word is permitted if this is required by the full element name. A maximum of 40 characters is allowed. The element description must be enclosed in single quote (') marks.

(12)

ELEMENT DESCRIPTION IS  
'THE ACRONYM OF AN ORGANIZATION'

5. ENTRY-SECURITY IS ENTRY-. [REQUIRED] This statement defines the security classification of the data element definition entries. Appendix E.1 identifies the valid security classifications for this entry.

(12)

ENTRY-SECURITY IS ENTRY-UNCLASSIFIED

6. DATA-SECURITY IS DATA-. [REQUIRED] This statement defines the security classification of the individual data element occurrences within the database. Appendix E.1 identifies the valid security classifications for this entry.

(12)

DATA-SECURITY IS DATA-UNCLASSIFIED

7. APPLICATION-SYSTEM IS. [REQUIRED] The subsystem name is entered. This statement provides a link between the data elements and the subsystem which they support. It is possible for a data element to support multiple subsystems. When this occurs, multiple occurrences of this statement should appear, one for each application. While the class name includes "system", the statement applies equally to systems and subsystems.

(12)

APPLICATION-SYSTEM IS NIMIS

8. SERVICE-SUPPORTED IS. The identifier of a service analysis supported by the data element is entered. This statement links data elements to service analyses. More than one service analysis may be identified.

(12)  
SERVICE-SUPPORTED IS S8120003

9. PICTURE IS. [REQUIRED] The user enters this information using the COBOL notation defined in Appendix B.3.

(12)  
PICTURE IS X(16)

10. USAGE IS. This statement identifies the internal representation of the data element within the database. Appendix B.4 identifies the standards for this entry. Most data elements are defined as DISPLAY usage.

(12)  
USAGE IS DISPLAY

11. VALUE IS. [REQUIRED] This statement defines the initial value which is to be present in the data description used by programs accessing the data element. The field is completed according to conventions defined in Appendix B.5.

(12)  
VALUE IS SPACES

12. STORE-VALIDATION IS. [REQUIRED] This entry defines the type of validation which is to be performed on incoming data being placed in the database when a record containing the data element is first stored. Enter a four-character identifier from Appendix E.8 which defines the validation to be performed.

(12)  
STORE-VALIDATION IS SSZA

13. MODIFY-VALIDATION IS. [REQUIRED] This entry defines the type of validation which is to be performed on incoming data which is to replace data values already resident in the database. Enter a four-character identifier from Appendix E.8 which defines the validation to be performed.

(12)

MODIFY-VALIDATION IS SMZB

14. ELEMENT DESIGNATOR IS. This statement may be used more than once, depending on what designators apply to the element. See Appendix E.10 for a list of valid designators.

(12)

ELEMENT DESIGNATOR IS DES-INDEX

15. JUSTIFY IS. This entry identifies non-standard justification of data within the element space. Normally data loaded to numeric fields is "right justified" which means it is loaded from the right-most position of the field to the left with leading zeros placed in unfilled left-most positions. Alphabetic or mixed data is loaded "left justified" which means that the data is loaded from the leftmost position of the field to the right with unused rightmost positions cleared to blanks. The value "ON" is entered if this justification is to be reversed. The clause is omitted if normal justification is desired.

(12)

JUSTIFY IS OFF

16. DECODE IS. This attribute defines special conversion features to be employed against the data element values. See Appendix E.6 for valid entries.

(12)

DECODE IS NO-DECODE



17. PRESENCE IS. [REQUIRED] This attribute defines the requirement for the field to be present in the database and in the IP's which load the database. See Appendix E.7. for valid entries.

(12)  
PRESENCE IS PRESENCE-REQUIRED

18. FILL IS. This attribute defines the use of a fill character to be used during justification of input and output transmission. The character is enclosed in single quotes. The default is space-fill.

(12)  
FILL IS ' ' (clause can be omitted)

19. ELEMENT DEFINITION IS. [REQUIRED] This entry contains a non-ADP definition of the data element and the manner in which it is entered into the database. The entry may be continued as desired by using comment/continuation forms. All definition lines must be entered into the data dictionary together. IT IS PARTICULARLY IMPORTANT THAT THIS DEFINITION BE UNDERSTANDABLE BY THE PERSON WHO NORMALLY ENTERS SUCH DATA INTO THE DATABASE. In fact, it is useful to ask the person entering the data to explain how the data is prepared and what it means and then use that explanation as the basis for this entry. The definition must be terminated by a single quote mark.

(7)      (12)  
ELEMENT DEFINITION IS  
- 'THE ELEMENT CONTAINS THE ABBREVIATED NAME OR  
- 'ACRONYM OF AN ORGANIZATION. IT IS LIMITED  
- 'TO A MAXIMUM OF 16 CHARACTERS AND MAY NOT  
- 'DUPLICATE ANOTHER ORGANIZATION ACRONYM ALREADY  
- 'STORED IN THE DATABASE'

20. DIA-REF-NO IS. This field identifies the relationship of the data element to a specific DIA reference. The purpose is to provide an effective cross-reference to data element standards.

(12)  
DIA-REF-NO IS X23004

21. REF-PUB IS. This field permits the user to define publications which reference the data element and add to its definition or understanding. Multiple entries may be coded where more than one reference exists.

(12)  
REF-PUB IS IDEAS

22. STANDARD IS. This field is used to define the existence of a standard definition of the data element. A list of applicable standards is provided in Appendix E.2.

(12)  
STANDARD IS IDEAS

23. SOURCE-DOC IS. The form number of the source document is entered, enclosed in single quote marks. If the form number is not available, enter the form name.

(12) SOURCE-DOC IS NONE

24. DATA-SUBJECT IS. [REQUIRED] This statement defines the data element as participating in one or more categories of data. Valid categories are defined in Appendix E.9. If the data element does not match any existing category, recommend an addition to the list to the DBA staff. Multiple word data subjects must be enclosed in single quotes.

(12)  
DATA-SUBJECT IS INSTALLATION

25. USER IS. [REQUIRED] This statement identifies the user organization/person which is responsible for definition of the data element. See Appendix A.3.

(12)  
USER IS 'NIPSSA51                    TOWNER'  
RESPONSIBLE FOR DEFINITION

26. ELEMENT NAME SYNONYM IS. It is possible that a subsystem will identify data elements which are already present within the database. The synonym clause is used when the same data element is used by multiple applications with different element names. The synonym feature permits a convenient renaming of a data element where it is used by multiple applications. The element physically occurs in only one place in the database but may be addressed by several names.

(12)  
ELEMENT NAME SYNONYM IS UNIT-DESIGNATOR

27. RANGE IS xxxxx THRU yyyyy. This version of the range statement permits the definition of two inclusive values which identify the valid values which may occur within the data element. Multiple statements may be prepared which define several ranges, including overlapping ranges.

(12)  
RANGE IS 'AAAAAAAAAAAAAAAA' THRU 'ZZZZZZZZZZZZZZZZ'  
(shown for example only-would not normally be used  
for this type of field)

28. RANGE IS. This statement is used to define specific valid values which the data element may contain within the database. Valid values entered are used to validate raw data submitted for database updating. An unlimited number of range value statements may be prepared.

(12)  
RANGE IS 'NIPSSA51                    '

NOTE: For both of the above range entries the range values must be enclosed in single quote marks. The length of the range value must be the same as the length of the associated data element defined in the PICTURE clause above. Numeric values must be right justified and zero filled to the left. For example: The data element is two digits long and has a permissible value of 5. The literal in the range clause is '05'. Where the second form of the range clause is used, the "xxxxx" value must be less than the "yyyyy" value.

29. COMMENTS. This entry is used to further define the data element. More technical information or conditional situations affecting the data may be described here. As many comment/definition continuation coding sheets as desired may be added as long as all comment lines are entered into the data dictionary at the same time. The last line of comments must be terminated by a single quote mark.

(12)

COMMENTS

- 'THE FIELD MAY BE A DIRECT ENTRY POINT INTO THE
- 'DATABASE. IT SHOULD BE EITHER A KEY OR AN
- 'INDEX'.

IT IS VERY IMPORTANT THAT A PERIOD BE PLACED AT THE END OF THE LAST STATEMENT IN THE DATA ELEMENT DEFINITION TO TELL THE DATA DICTIONARY THAT THE DATA ELEMENT DEFINITION IS COMPLETE.

ADD ELEMENT NAME IS ORGANIZ-ACRONYM  
 PREPARED BY LET  
 SAME AS ELEMENT ORGAN-ACRONYM  
 ELEMENT DESCRIPTION IS  
 'THE ACRONYM OF AN ORGANIZATION'  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-UNCLASSIFIED  
 APPLICATION-SYSTEM IS NIMIS  
 SERVICE-SUPPORTED IS S8120003  
 PICTURE IS X(16)  
 USAGE IS DISPLAY  
 VALUE IS SPACES  
 STORE-VALIDATION IS SSZA  
 MODIFY-VALIDATION IS SMZE  
 ELEMENT DESIGNATOR IS DES-INDEX  
 JUSTIFY IS OFF  
 DECODE IS NO-DECODE  
 PRESENCE IS PRESENCE-REQUIRED  
 FILL IS ' '  
 ELEMENT DEFINITION IS  
 - 'THE ELEMENT CONTAINS THE ABBREVIATED NAME OR  
 - 'ACRONYM OF AN ORGANIZATION. IT IS LIMITED  
 - 'TO A MAXIMUM OF 16 CHARACTERS AND MAY NOT  
 - 'DUPLICATE ANOTHER ORGANIZATION ACRONYM ALREADY  
 - 'STORED IN THE DATABASE'  
 DIA-REF-NO IS X23004  
 REF-PUB IS IDEAS  
 STANDARD IS IDEAS  
 SOURCE-DOC IS NONE  
 DATA-SUBJECT IS INSTALLATION  
 USER IS 'NIPSSA51 TOWNER'  
 RESPONSIBLE FOR DEFINITION  
 ELEMENT NAME SYNONYM IS UNIT-DESIGNATOR  
 RANGE IS 'NIPSSA51 '  
 COMMENTS  
 - 'THE FIELD MAY BE A DIRECT ENTRY POINT INTO THE  
 - 'DATABASE. IT SHOULD BE EITHER A KEY OR AN  
 - 'INDEX'.

FIGURE A.6.1.



## APPENDIX A.7

### IDD GROUP DATA ELEMENT DEFINITION ENTRY

Data elements within an application are normally related in some manner. This relationship may be very loose, with the only connection being the application as a whole. Some elements, however, are very closely related. For example: Month, day, and year are the related parts of a date. When this occurs, it is desirable to link those elements within the IDD so that they cannot be inadvertently separated during the design process. Grouping also improves the organization of the database and the usability of the stored data.

The easiest way to begin the grouping of data elements which were defined in Appendix A.6 is to physically group the coding sheets for related elements. Each group should be clipped together to prevent confusion. Once the groupings have been made, each group should be reviewed to determine the ordering of elements within the group. While this order can be made by any criteria, the following order is suggested as a general guideline:

1. Key elements which determine an identity or ordering of the group, particularly if the group is repeated within the record occurrence.
2. Any other elements which make the group unique.
3. Any elements which will be used as entry pointers into the database should be placed as close to the front of the group as possible.
4. Dates.
5. Fixed length numeric fields.

6. Fixed length alphanumeric fields.

7. Textual fields.

The effect of this ordering approach is to place fields which control the group at the beginning, followed by fixed fields which are likely to be completely filled by data, and ended by fields which may not be completely data filled. This organization optimizes the data compression feature used by the DBMS.

Figure A.7.1. illustrates the statements required to group data elements together.

#### DATA ELEMENT GROUP DEFINITION

1. ADD ELEMENT NAME IS. [REQUIRED] This entry assigns the name by which the grouped elements may be accessed as a whole. The name must conform to data name conventions defined in Appendix B. As with elementary data elements, it may be easier to define the full name of the group (step 3 below) and then define the shorter group name.

(8)

ADD ELEMENT NAME IS DATE-OF-BIRTH

2. PREPARED BY. [REQUIRED] The initials of the preparer are entered.

(12)

PREPARED BY LET

3. ELEMENT DESCRIPTION IS. [REQUIRED] This statement mits a more detailed expansion of the name of the group.

(12)

ELEMENT DESCRIPTION IS  
'DATE ON WHICH A PERSON WAS BORN'

4. ENTRY-SECURITY IS ENTRY-. [REQUIRED] This statement identifies the security classification of the description of the data element group. Valid classifications are defined in Appendix E.1.

(12)

ENTRY-SECURITY IS ENTRY-UNCLASSIFIED

5. DATA-SECURITY IS DATA-. [REQUIRED] This statement identifies the security classification of the contents of the data element group within the database. Valid security classifications are defined in Appendix E.1.

(12)

DATA-SECURITY IS DATA-UNCLASSIFIED

NOTE: The data security entry defines the aggregate classification of the grouped elements. The defined classification must be at least equal to the highest classification of any of the individual elements. IN SOME CASES, THE GROUPING OF DATA ELEMENTS MAY ESTABLISH A SECURITY CLASSIFICATION HIGHER THAN THAT OF ANY INDIVIDUAL ELEMENT.

6. ELEMENT DEFINITION IS. [REQUIRED] This statement permits a detailed description of the data element group. THE DESCRIPTION MUST BE IN NON-ADP TERMS which can be readily understood by the person who will prepare the data for entry into the database.

(12)

ELEMENT DEFINITION IS

'THE DATE ON WHICH A PERSON WAS BORN'

7. ELEMENT DESIGNATOR IS. This statement permits assigning special designation identifiers to the group element. More than one designator may be assigned if applicable to the element. See Appendix E.10 for a list of valid designators.

(12)

ELEMENT DESIGNATOR IS DES-DATE

8. DECODE IS. This attribute defines special conversion features to be employed against data element values. See Appendix E.6 for valid entries. This entry is used only when a decode applies to the entire element group, such as dates.

(12)  
DECODE IS DATE-EDIT

9. DIA-REF-NO IS. This field permits associating a DIA reference to the data element group.

(12)  
DIA-REF-NO IS X22119

10. REF-PUB IS. This field identifies reference publications which are associated with the data element group. Multiple entries are permitted.

(12)  
REF-PUB IS IDEAS

11. STANDARD IS. This field identifies a specified ADP standard which is related to the data element group. More than one entry is permitted. Appendix E.2 identifies applicable standards publications.

(12)  
STANDARD IS IDEAS

12. SUBORDINATE ELEMENTS ARE. [REQUIRED] These statements define the individual data elements which comprise the group. The order in which the elements are placed in the group is determined by the order in which they are entered in this statement. The coding form permits defining multiple occurring fields by coding 'OCCURS' in the second section of the clause line, followed by the number of occurrences in the third section of the clause line.

(12)  
SUBORDINATE ELEMENTS ARE  
YR-OF-BIRTH  
MO-OF-BIRTH  
DY-OF-BIRTH.

13. COMMENTS. The comments entry may be coded in the same manner as the comments entry described in Appendix A.6 for elementary data elements. The comments entry is specifically for the use of ADP-oriented information which is beneficial to the user and system designer. Use additional comment/definition continuation forms as required to completely define the element group. The comment clause must be terminated by a single quote mark.

(12)

COMMENTS

- 'A STANDARD DATE GROUP'.

14. The entry must be terminated by a period.

ADD ELEMENT NAME IS 'DATE-OF-BIRTH'  
PREPARED BY LET  
ELEMENT DESCRIPTION IS  
'DATE ON WHICH A PERSON WAS BORN'  
ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
DATA-SECURITY IS DATA-UNCLASSIFIED  
ELEMENT DEFINITION IS  
'THE DATE ON WHICH A PERSON WAS BORN'  
ELEMENT DESIGNATOR IS DES-DATE  
DECODE IS DATE-EDIT  
DIA-REF-NO IS X22119  
REF-PUB IS IDEAS  
STANDARD IS IDEAS  
SUBORDINATE ELEMENTS ARE  
YR-OF-BIRTH  
MO-OF-BIRTH  
DY-OF-BIRTH  
COMMENTS  
'A STANDARD DATE GROUP'.

Figure A.7.1.



## APPENDIX A.8

### IDD ENTITY DATA DEFINITION ENTRY

The final act of associating data elements and element groups is to organize these elements and groups into the major groups comprising the data entity definition. The data entity represents the creation of a database record when data is stored in the database itself.

The element/element group associations which combine to form a data entity are:

1. The fixed information which is present in all database records. This information consists of the security classification, releasability code, date on which the record occurrence was last modified, the database identifier, logical record type code, and the date on which the record occurrence was originally stored. This section is identified by the entity name prefixed by "FI-". The section is always at the beginning of the entity.
2. The identifier. This association consists of those elements and groups which together establish each occurrence of the data entity within the database as unique. The element name for this association is the name of the entity prefixed by "ID-".
3. The body of the entity. This association contains the remaining data element and element groups associated with the database entity. The element name for this association is the name of the entity prefixed by "DA".

The format of data element group definition described in Appendix A.7 is also used in this appendix.

ADD ELEMENT NAME IS FI-ORGANIZATION  
PREPARED BY LET  
ELEMENT DESCRIPTION IS  
'FIXED PORTION OF ORGANIZATION RECORD'  
ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
DATA-SECURITY IS DATA-UNCLASSIFIED.  
SUBORDINATE ELEMENTS ARE  
    CLASS-1000  
    HANDL-1000  
    DATE-MOD-1000  
    DBID-1000  
    LOG-TYPE-1000  
    DATE-STORED-1000.

ADD ELEMENT NAME IS ID-ORGANIZATION  
PREPARED BY LET  
ELEMENT DESCRIPTION IS  
'IDENTIFIER OF ORGANIZATION RECORD'  
ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
DATA-SECURITY IS DATA-UNCLASSIFIED  
ELEMENT DESIGNATOR IS DES-KEY.  
SUBORDINATE ELEMENTS ARE  
    ORGAN-ACRONYM.

ADD ELEMENT NAME IS DA-ORGANIZATION  
PREPARED BY LET  
ELEMENT DESCRIPTION IS  
'DATA FOR ORGANIZATION RECORD'  
ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
DATA-SECURITY IS DATA-UNCLASSIFIED.  
SUBORDINATE ELEMENTS ARE  
    NAME-ORGANIZATION  
    ORGAN-CODE  
    ADDR-ORGAN  
    CITY-ORGAN  
    ZIP-ORGAN.

Figure A.8.1

## APPENDIX A.9

### IDD USER DATABASE AUTHORIZATION

Each user must be overtly associated with those elements within the database which may be accessed by the user. Without such an authorization, input and retrieval errors will occur during processing. While a user may be permitted access to certain database records, access to data elements within the records is dependent upon specific authorization.

The IDD authorization entries define four interface modes which the user may assume: access; store; modify, and deletion of data elements.

### USER DATABASE AUTHORIZATION

1. MODIFY ELEMENT NAME IS. [REQUIRED] This clause identifies the data element which the user will be permitted to act upon.

(8)

MODIFY ELEMENT NAME IS YR-OF-BIRTH

2. INCLUDE USER IS. [REQUIRED] This clause, when user, permits the user to retrieve the data element from the database. If one of the following three options of the INCLUDE USER IS clause is chosen, this statement is not necessary. Retrieval authority is automatically granted as part of the three options.

(12)

INCLUDE USER IS 'NIPSSA51 TOWNER'

3. INCLUDE USER IS xxxxx RESPONSIBLE FOR CREATION. This clause, when used, permits the user to store the data element value in the database.

(12)

INCLUDE USER IS 'NIPSSA51 TOWNER'  
RESPONSIBLE FOR CREATION

4. INCLUDE USER IS xxxxx RESPONSIBLE FOR UPDATE. This clause, when used, permits the user to modify the data element values in the database.

```
(12)
INCLUDE USER IS 'NIPSSA51      TOWNER'
RESPONSIBLE FOR UPDATE
```

5. INCLUDE USER IS xxxxx RESPONSIBLE FOR DELETION. This clause, when used, permits the user to delete the data element from the database.

```
(12)
INCLUDE USER IS 'NIPSSA51      TOWNER'
RESPONSIBLE FOR DELETION
```

NOTE: Any combination of these entries may be used. A period must follow the last entry. If the user is permitted all three update options (creation, update, and deletion), a single clause "INCLUDE USER IS xxxxx RESPONSIBLE" may be used in place of the three individual entries.

## APPENDIX A.10

### SERVICE ANALYSIS PREPARATION

The service analysis approach applied in this Guide is an adaptation of the technique promoted by Leo Cohen for defining and prioritizing the services which are to be supplied by the database application. IDD does not directly support a service analysis function. However, the program entry within the data dictionary is very similar to a service function.

A number of attributes have been added to support the service analysis. Where the service analysis describes output reports, the procedure described in Appendix A.11 should be followed. It is not necessary to describe both a service analysis entry and a report entry for the same service. Figure A.10.1 illustrates the statements required to complete one service analysis entry.

#### IDD SERVICE ANALYSIS ENTRY

1. ADD PROGRAM NAME IS. [REQUIRED] The program name is limited to 8 bytes. Its format is:

- a. The character "S",
- b. The NIPSSA project number of the application, 4 characters.
- c. sequential number of the service, ranging from 001 through 999.

(8)

ADD PROGRAM NAME IS S8120001

2. PREPARED BY. [REQUIRED] The initials of the analyst preparing the service analysis are entered.

(12)

PREPARED BY 'LET'



3. PROGRAM DESCRIPTION IS. [REQUIRED] This 40-character (maximum) literal contains the title of the service function. It is enclosed in single quote marks.

(12)

PROGRAM DESCRIPTION IS  
'PROJECTS BY REQUESTING ORGAN RPT'

4. ENTRY-SECURITY IS ENTRY-. [REQUIRED] The security classification of the service analysis description is used. Appendix E.1 defines valid classifications.

(12)

ENTRY-SECURITY IS ENTRY-UNCLASSIFIED

5. DATA-SECURITY IS DATA-. [REQUIRED] The security classification of the data which the service function must utilize is entered. See Appendix E.1 for valid classification.

(12)

DATA-SECURITY IS DATA-UNCLASSIFIED

6. OUTPUT-SECURITY IS OUT-. The security classification of the output, if any, of the service function is entered. See Appendix E.1 for valid classifications.

(12)

OUTPUT-SECURITY IS OUTPUT-UNCLASSIFIED

7. WITHIN SUBSYSTEM. [REQUIRED] The name of the subsystem supported by the service function is entered.

(12)

WITHIN SUBSYSTEM NIMIS

8. OUTPUT-MODE IS. Appendix E.16 identifies the valid attributes when the service function is output oriented.

(12)

OUTPUT-MODE IS REPORT-MODE

9. FREQUENCY IS. [REQUIRED] Appendix E.12 defines the attributes which describe the frequency of use of the service. Multiple entries are possible.

(12)

FREQUENCY IS ADHOC

10. SERVICE-PRIORITY IS. [REQUIRED] Appendix E.13 defines the attributes which describe the priority of the service with regard to other functions being proposed.

(12)

SERVICE-PRIORITY IS NORMAL-PRIORITY

11. RESPONSE-TIME IS. [REQUIRED] Appendix E.14 defines the attributes which describe the response time required for the service when it is available.

(12)

RESPONSE-TIME IS OVERNIGHT-RESPONSE

12. MODE IS. This clause is "BATCH" if the service will be run in a batch environment; it is "SHADOW" if it will be run on-line.

(12)

MODE IS BATCH

13. POINT-OF-CONTACT IS. [REQUIRED] This clause contains the name of the user person who is the primary point of contact for the service function. The name is enclosed in single quote marks.

(12)

POINT-OF-CONTACT IS 'JOHN R JONES'

14. POC-PHONE IS. [REQUIRED] This clause contains the telephone number of the user point of contact defined above. The number is enclosed in single quotes.

(12)

POC-PHONE IS '325-0760'

15. USER-ORGANIZATION IS. [REQUIRED] This clause contains the acronym of the user organization supported by the service. The acronym must be enclosed in single quotes and must be identical to the user name identified in Appendices 3 or 5 as the service requestor.

(12)

USER-ORGANIZATION IS 'NIPSSA51'

16. SERVICE-HISTORY IS. [REQUIRED] This clause contains the history of the service provided to the user. Appendix E.15 contains valid codes.

(12)

SERVICE-HISTORY IS EXISTING-SERVICE

17. COMMENTS. [REQUIRED] The comments section will explicitly describe the service to be performed, its purpose, intended audience, and anything else about the service useful to development. This section will also identify any special conditions or features which the analyst needs to know.

(12)

COMMENTS

'THE REPORT . . . . .'

ADD PROGRAM NAME IS S8120001  
PREPARED BY 'LET'  
PROGRAM DESCRIPTION IS  
'PROJECTS BY REQUESTING ORGAN RPT'  
ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
DATA-SECURITY IS DATA-UNCLASSIFIED  
OUTPUT-SECURITY IS OUTPUT-UNCLASSIFIED  
WITHIN SUBSYSTEM NIMIS  
OUTPUT-MODE IS REPORT-MODE  
FREQUENCY IS ADHOC  
SERVICE-PRIORITY IS NORMAL-PRIORITY  
RESPONSE-TIME IS OVERNIGHT-RESPONSE  
MODE IS BATCH  
POINT-OF-CONTACT IS 'JOHN R JONES'  
POC-PHONE IS '325-0760'  
USER-ORGANIZATION IS 'NIPSSA51'  
SERVICE-HISTORY IS EXISTING-SERVICE  
COMMENTS  
'THE REPORT . . . . '.

Figure A.10.1

## APPENDIX A.11

### IDD REPORT PROGRAM ENTRY

This appendix acts as a logical extension to Appendix A.10. The service analysis for reports, either batch or on-line, is prepared. The additional attributes described in this appendix are added immediately before the COMMENTS.

1. LANGUAGE IS. The language to be used for the report is entered from the list in Appendix E.3.

(12)  
LANGUAGE IS CULPRIT

2. DISTRIBUTION IS. The number of copies of a printed report is entered, enclosed in single quote marks.

(12)  
DISTRIBUTION IS '2'

3. OUTPUT-FORM IS. The output form type to be used is entered from Appendix E.4, enclosed in single quote marks.

(12)  
OUTPUT-FORM IS '8 1/2 X 14'

4. AVERAGE-SIZE IS. The average number of pages of a printed report is entered, enclosed in single quote marks.

(12)  
AVERAGE-SIZE IS '25'

5. LEVELS-OF-TOTALS IS. The number of levels of arithmetic totals is entered, enclosed in single quote marks.

(12)  
LEVELS-OF-TOTALS IS '2'



6. DATE-RANGE IS. "INCLUSIVE" is entered if the report covers data within two inclusive dates. "THRU" is entered if the report covers all data up to and including the date supplied. "FOLLOWING" is used if the report covers all data subsequent to the date supplied. "SPECIFIC" is entered if the report covers data only for a specific date value supplied.

(12)

DATE-RANGE IS 'THRU'

7. COVER-PAGE IS. "STANDARD" is entered if the report utilizes the standard NIPSSA cover page. "SPECIAL" is used if a special cover page is required. Both conditions may exist.

(12)

COVER-PAGE IS STANDARD

## APPENDIX A.12

### BATCH INPUT PROCESSING (IP) SPECIFICATION DEFINITION

This appendix describes the procedures for preparing specifications for input processing (IP) program modules which will be created using the batch IP module generator program.

Each IP module is designed and created as an independent entity. The philosophy employed is that all input processing is transaction driven, each transaction independent of those preceding and following. This approach insures that incoming data can be processed without regard to its environment.

Each IP module is independent of the input medium. The module expects its data in a particular format and processes the data according to the specification.

Each IP program consists of four general groups of COBOL source statements:

- a. Generated statements which do not vary from IP to IP. These statements are loaded into the generated program from a PSL member.
- b. Generated statements which are tailored to each individual IP, such as program name, version, level, etc.
- c. Generated statements based on parameters supplied by the analyst or extracted from information within the data dictionary.
- d. Inserted COBOL source statements supplied by the analysts.

A combination of these four statement types are interspersed throughout the created COBOL program. The order and location of statements is controlled by the parameter identifier and the sequence of individual parameters.

Approximately two-thirds of the IP program consists of standard generated statements which do not vary between IP's. The analyst cannot modify any of these statements nor alter the order of their insertion into the source program. This has the dual effect of reducing the frequency of error in program definition and enhancing the security of the software updating the database.

Batch IP program creation consists of two separate processes:

- a. Definition of the IP format. This process utilizes the record definition format. The entries are tailored to the IP format definition.
- b. Preparation of IP definition parameters. Parameters describing the functions to be performed by the generated IP programs are prepared.

#### IP FORMAT DEFINITION FOR IDD

Each IP format is composed of an identifier (IP-IDENT) and a series of data elements which the IP will process. An IDD record definition is prepared which defines the format for use by the IP.

The order of element arrangement in the IP format is not critical to processing. It is useful, as a general guideline to:

1. Place elements necessary to establish currency to the target record at the left (beginning) of the card following IP-IDENT. Where multiple currencies are to be established, arrange the elements from left to right in the order
2. Place required elements, those which must have data entered, next. This avoids having data entry operators space or skip over optional fields to reach required fields. 1. ADD RECORD NAME IS IP-. This statement defines the IP format as a record description within the data dictionary.

1. ADD RECORD NAME IS IP-. This statement defines the IP format as a record description within the data dictionary.

(8)

ADD RECORD NAME IS IP-AATS

2. PREPARED BY. The initials of the preparer are entered.

(12)

PREPARED BY LET

3. RECORD DESCRIPTION IS. This statement is used to identify the IP and its record entity, e.g., STORE NIM-PROJECT. The description is bounded by single quote marks.

(12)

RECORD DESCRIPTION IS 'STORE NIMIS PROJECT RECORD'

4. RECORD STORAGE IS IP. This statement identifies the record type as an IP.

5. ENTRY-SECURITY IS ENTRY-. The security classification of the IP description is entered. See Appendix E.1 for valid security classifications.

(12)

ENTRY-SECURITY IS ENTRY-UNCLASSIFIED

6. DATA-SECURITY IS DATA-. The security classification of the data which the IP will pass to the database is defined. See Appendix E.1 for valid security classifications.

(12)

DATA-SECURITY IS DATA-UNCLASSIFIED

7. MODE IS BATCH. The statement identifies the IP as a batch module.

8. COMMENTS. The comments section will provide the basis for the users guide description of the IP and its functions. The comments, therefore, must be in non-ADP terms understandable by functional user personnel.

AD-A069 868

NAVAL INTELLIGENCE PROCESSING SYSTEMS SUPPORT ACTIVIT--ETC F/G 9/2  
INTEGRATED DATABASE DEVELOPMENT AND DESIGN GUIDE. VERSION 2.0.(U)  
MAY 79

UNCLASSIFIED

NIPSSA-UM-8000/2.0-5/79

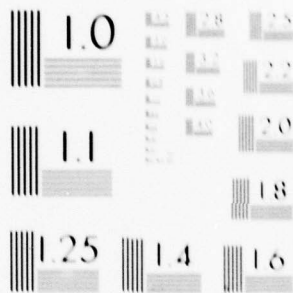
NL

2 OF 7

AD  
A069868







MICROCOPY RESOLUTION TEST CHART  
 NATIONAL BUREAU OF STANDARDS-1963-A

Record element statements are prepared for each element which appears in the IP format. The elements are entered in the same order as they will appear, from beginning to end, in the format.

The first record element in any batch IP definition is IP-IDENT. This element is a group combining IP-FORMAT and IP-OPTION. This element must be the first one in every batch IP definition.

(12)  
RECORD ELEMENT IS IP-IDENT.

9. RECORD ELEMENT IS. This statement identifies the element or element group to be included in the IP format.

(12)  
RECORD ELEMENT IS DATE-OF-BIRTH.

ADD RECORD NAME IS IP-AATS  
PREPARED BY LET  
RECORD DESCRIPTION IS 'STORE NIMIS PROJECT RECORD'  
RECORD STORAGE IS IP  
ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
DATA-SECURITY IS DATA-UNCLASSIFIED  
MODE IS BATCH  
COMMENTS  
'THE IP IS THE PRIMARY IP FOR THE NIMIS PROJECT  
'RECORD'.  
- RECORD ELEMENT IS IP-IDENT.  
RECORD ELEMENT IS DATE-OF-BIRTH.

Figure A12.1

## IP PARAMETER DECK PREPARATION

Parameter descriptions apply to Version 1 of the batch IP generator program ZIPX. (Figure A12.4 illustrates a complete parameter deck.)

1. The CTLP parameter control card is required and must be the first statement (positions 1-4) in the parameter deck. It provides information specific to the IP being generated.
2. THE IP IDENTIFIER. (6-8) This identifier is assigned by the DBA staff. It consists of a three-character alphabetic code which establishes each unique IP format.
3. IP OPTION. (9) This field defines the primary function of the IP. Valid codes are:
  - a. "S" - Store a new occurrence of a data entity within the database.
  - b. "M" - Modify a data entity occurrence already stored in the database.
  - c. "D" - Delete a data entity occurrence currently stored in the database.
4. SUPPORTING SUBSCHEMA. (11-14) This field defines the name of the subschema supporting the IP. The "SS" prefix of the subschema is omitted.
5. SCHEMA. (16-23) This field contains the name of the schema supporting the IP.
6. VERSION. (30-31) This field identifies the revision version of the IP. It may contain numeric values from 01 through 99.
7. LEVEL. (33-34) This field identifies the minor revision level of the IP. It may contain numeric values from 01 through 99.
8. TARGET RECORD NAME. (36-51) This field defines the schema name of the database record type to be processed by the IP.
9. TARGET RECORD IDENTIFIER. (53-56) The schema numeric identifier of the database record type to be processed by the

IP is entered.

CTLTP AATS NIMI NIM				01 01 ORGANIZATION 1000				
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)

Figure A12.2

IP modules are designed and programmed using structured programming techniques which grouped similar functions to achieve more accurate processing. A facility of the COBOL compiler permits definition of common program source statements into source library "books".

These books may be modified when they are requested by the program, tailoring the statements to the specific data element being serviced. This approach insures consistent common language logic and eliminates large volumes of statement preparation. The IP generator program makes extensive use of this source book feature.

IP programs are broken into three main sections (Figure A12.3):

1. A section which is entered when the IP is initially loaded into memory and executed. This section performs IDMS bind and ready functions and verifies the user authorization to use the IP.
2. A section which is entered each time data belonging to the IP is sensed by the control program. This section performs all data element validation and database maintenance.
3. A section which is entered when all data belonging to the IP has been exhausted. The section performs IDMS finish functions after updating processing statistics.

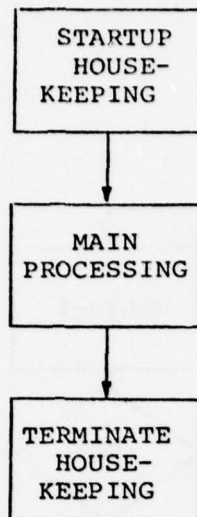


Figure A.12.3

The main processing section of the IP is further broken into three subsections (Figure A12.4):

1. A section which performs data element validation and data transfer functions which can be accomplished without using IDMS(VALID-1). Parameters which apply to this section begin with a control code of "1".
2. A section which performs data element validation and transfer functions which require access to the database. Parameters which apply to this section begin with a control code of "2" (VALID-2).
3. A section which performs all IDMS database support functions and associated activity to complete IP processing. This section provides the an analyst with a great deal of flexibility. Parameters which apply to this section begin with control codes of "3", "5", and "6" (CURRENCY-CTL).



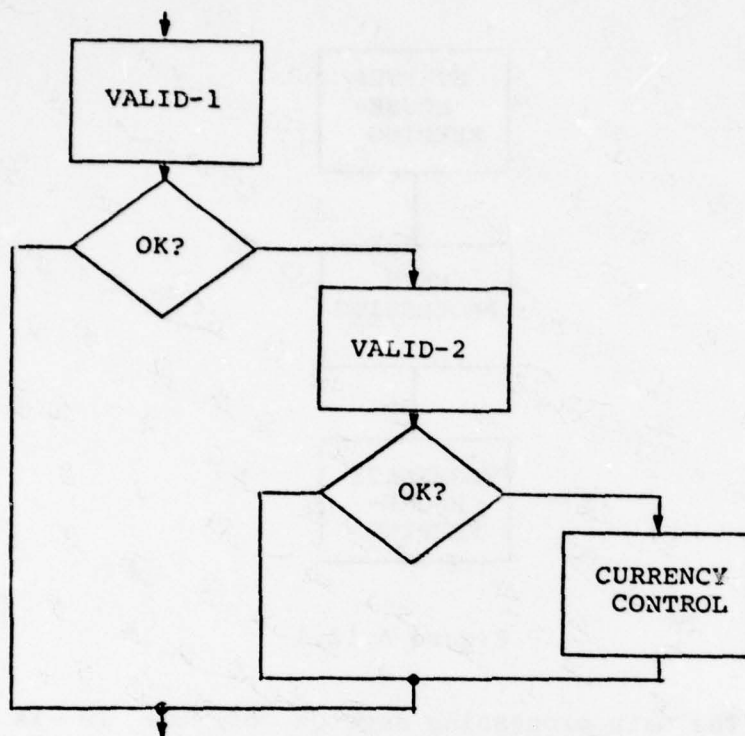


Figure A.12.4

#### WORK FIELD DEFINITION

The basic IP provides two subscripts, R and RX, for use by program logic. A single 60-character alphanumeric work field, IP-WORK, is provided for intermediate storage. Where additional fields are required, the analyst must define the required fields.

Work fields are defined using the "A" parameter statement.

1. "A" identifier. This identifier must be in the first position of the parameter, followed by a comma (delimiter).
2. COBOL Data Division Statement. A standard COBOL Data Division statement is prepared following COBOL columnar alignment conventions. Statements are transferred to the generated source program exactly as entered except that '01-01' is placed in positions 1-5 of the source lines.

A, 01 HOLD-AREA PIC X(8).

(1)            (2)

#### IP FORMAT DEFINITION

The current version of ZIPX requires that the analyst define the fields within the IP format explicitly. These parameters are converted to COBOL source statements within the Linkage Section of the generated program. The format definition identifies the data fields from the 80-character block passed to the IP from the control program. Figure A12.5 illustrates a complete IP parameter group.

1. "C" Identifier. This identifier must be in the first position of the parameter, followed by a comma.

2. (Field Name). The name of the data element in the IP format is entered. This name may not exceed 24 characters including a subscript, if required. The field name is followed by a comma. "FILLER" may be used for undefined fields.

3. Field Length. The number of bytes (format positions) required by the field is entered.

C,AZA-IDENT,10

(1)   (2)        (3)

Figure A12.5

#### DATA ELEMENT VALIDATION AND TRANSFER

Data element validation and transfer is performed through source library books. Depending on the nature of the element and its use by the IP, an element may be validated and transferred from the input area to the database record area in any one of the three processing subsections described above. Figure A12.6 illustrates a validation and transfer entry.

The subsection in which the validation and transfer occurs is determined by the parameter identifier:

1. When the parameter identifier is "1", validation and transfer occurs in the first subsection (no database interaction required).
2. When the parameter identifier is "2", validation and transfer occurs in the second subsection (database interaction required to verify data values).
3. When the parameter identifier is "3", validation and transfer occurs in the third subsection (database currency and maintenance).

Determining where to perform validation and transfer is very important to the successful operation of the IP. Subsection one is used:

1. For all data elements which do not require table lookup for data verification when the IP option is STORE or DELETE.
2. For those data elements which establish currency when the IP option is MODIFY.

Subsection two is used:

1. For all data elements which require table lookup for data verification when the IP option is STORE or DELETE.
2. For data elements which require table lookup for data verification AND are required for currency when the IP option is MODIFY.

Subsection three is used:

1. For all data elements whose values will replace existing values in the database. This validation must take place following the reading of the data entity occurrence to be modified.

2. For any special transfer of data desired to accomplish the IP function.

The format in the validation and transfer parameter is the same whether used in subsections 1, 2, or 3:

1. n Subsection Identifier. This field, followed by a comma, defines the subsection in which the COBOL source will be entered. It begins in column 1 of the parameter card.
2. Book Identifier. This field identifies the source library book which will be used to validate and transfer the data element contents. A comma follows the identifier.
3. Field Position. This field identifies the relative field location on the IP line of the data element. This information is used on error messages to assist the user in locating which element is the subject of a message. Fields are numbered from left to right, with the IP format being defined as field 01.
4. (field name). This field defines the name of the source element defined by "C" parameters above.
5. (element name). This field defines the data element name within the data entity occurrence where the verified data is to be transferred.
6. Optional Field A. This field contains the lowest inclusive range value for those source books using range validation. It contains the name of the database element name, enclosed in single quote marks, for those books which invoke table validation.
7. Optional Field B. This field contains the highest inclusive range value for those source books using range validation.

1,SSZA,03,AAA-IDENT,NIM-PROJ-ID,aa,bb

(1) (2) (3) (4) (5) (6) (7)

Figure A12.6



## IDMS DML COMMANDS

IDMS DML commands are defined through use of the "5" parameter. Figure A12.7 illustrates the DML parameter. The parameter is a combination of parametric values and a COBOL DML command:

1. "5" Parameter Identifier. The parameter identifier must appear in the first position of the line, followed by a comma.

2. Source Book Identifier. The source book which will perform the validation of DML results is entered. Valid books are:

- a. OBTA - obtain a database record occurrence.
- b. FIND - locate and establish currency with a database record occurrence but do not bring it into the programs work area.
- c. STOR - store a new database record occurrence and issue a message defining the results of the action.
- d. STRN - store a new database record occurrence and issue a message only if the store attempt was unsuccessful.
- e. MODI - modify an existing database record occurrence and issue a message defining the results of the action.
- f. MODN - modify an existing database record occurrence and issue a message if the attempt was not successful.
- g. ERAS - erase/delete an existing database record occurrence from the database and issue a message defining the results of the action.

3. Field Location. This field defines the location of the field on the IP line which is the primary source of information for the DML command.

4. DML Command. The actual DML command to be executed is



coded in COBOL format including a period at the end of the command.

5,STOR,04,STORE NIM-PROJECT.

(1) (2) (3) (4)

Figure A12.7

#### COBOL PROCEDURE SECTIONS

It is possible that certain IP's will require processing that is too complex to be handled effectively by in-line code. When this occurs, a "6" statement can be written to perform other self-contained procedure logic. This logic is loaded to the end of the generated program as COBOL sections. This parameter assumes that all COBOL statements begin in position 12 of the source card. If it is desired to begin the statement in position 16, enter 4 spaces between the comma and the first word of the COBOL statement.

6,PERFORM SPECIAL-HANDLING.

The generator program recognizes this procedure code as "7" parameters. With the exception of the "7," in the first two positions of the line, the parameter is a mirror of standard COBOL procedure source statements. All COBOL columnar alignment conventions are followed.

These guidelines must be followed for all type "7" parameters:

1. All logic will be self-contained without use of GO TO statements exiting beyond the routine.
2. COBOL sections will be used as objects of PERFORM statements. Paragraphs may be used within sections.
3. Each section will be terminated by an EXIT statement.

4. Each section will be preceded by comments describing the purpose and functioning of the section. Additional comments will be interspersed throughout the logic of the section describing each logical operation as it occurs.
5. Structured programming techniques will be followed utilizing the three coding forms of simple sequence, do while (until), and if then else. Comments will precede each occurrence of one of these forms except that one set of comments may be used for a series of similar simple sequences such as moves.
6. Visual formatting will be used to improve the readability of the source statements. These formatting procedures will be followed:
  - a. Each section will begin on a new page by using the EJECT statement.
  - b. Each comment will be preceded by two blank lines and followed by one blank line, utilizing the SKIPn statement.
  - c. The text of all comments will begin in column 20 and will be preceded and followed by a line of asterisks to emphasize the comments and prevent comments from being confused with procedure code.
  - d. Elementary procedure statements will begin in column 12. Statements subsidiary to a preceding statement or a continuation of a preceding statement will be indented 4 columns.
  - e. Section and paragraph names will not exceed 16 characters and will be as descriptive as possible.

#### PARAMETER DECK TERMINATION

The parameter deck is terminated by a statement containing a "9" in the first position. Omission of this statement will cause invalid results.

```
CTLP AATS NIMI NIM      01 01 ORGANIZATION 1000
A,      01 HOLD-AREA    PIC X(8).
C,AZA-IDENT,10
C,AZA-ADDR,25
1,SSZA,03,AZA-IDENT,NIM-PROJ-ID
1,SSZA,04,AZA-ADDR,ADDR-NIM
6,MOVE CURRENT-DATE TO HOLD-AREA.
6,      MOVE HOLD-AREA TO DATE-STORED-1000.
5,STOR,03,STORE ORGANIZATION.
9,
```

Figure A12.8

## APPENDIX A.14

### ON-LINE INPUT PROCESSING (IP) DEFINITIONS

This appendix describes the procedures for preparing specifications for input processing (IP) program modules which will be created using the on-line IP module generator program.

Each IP module is designed and created as an independent entity. The philosophy employed is that all input processing is transaction driven, each transaction independent of those preceding and following. This approach insures that incoming data can be processed without regard for its environment.

Each IP module is independent of the originating terminal. The module expects its data in a particular format and processes the data according to the specification.

Each IP program consists of four general groups of COBOL source statements;;

- a. Generated statements which do not vary from IP to IP. These statements are loaded into the generated program from a PSL member. Depending on the processing option (store, modify, or delete) and the currency path, different standard statements be copied.
- b. Generated statements which are tailored to each individual IP, such as program name, version, level, buffers, etc.
- c. Generated statements based on parameters supplied by the analyst or extracted from information within the data dictionary.
- d. Inserted COBOL source statements supplied by the analysts.

A combination of these four statement types are interspersed throughout the created COBOL program. The order and location of statements is controlled by the parameter identifiers and the sequence of individual parameters.

Approximately two-thirds of the IP program consists

of standard generated statements which do not vary between IP's. The analyst cannot modify any of these statements nor alter the order of their insertion into the source program. This has the dual effect of reducing the frequency of error

in program definition and enhancing the security of the software updating the database.

On-line IP program creation consists of five separate processes:

- a. Definition of the database currency requirements for the IP. These parameters are identified by a "A" and "B".
- b. Definition of the IP screen display. This process utilizes a special set of input parameters to the program generator. All screen definition parameters are begun with a "D", delimited by a "C" starting parameter and a "E" terminating parameter.
- c. Definition of the elements which are to be processed by the IP. A set of input parameters beginning with "H" identify these elements.
- d. Definition of logically related (follow on) IP displays. A set of input parameters which begin with "M" identify these processing options.
- e. Processing of the elements and records associated with the IP. The parameter descriptions defined for batch IP's in Appendix A.12 are used with minor changes to define the processing functions.

While the five functions are defined separately for understanding and convenience, they are highly interrelated. The order of their presentation corresponds with the order they appear in the parameter deck. From a design view, however, the process will typically flow in this order:

- a. Definition of the purpose of the IP screen (store, modify, delete) and the principal



user of the screen.

- b. Definition of the data elements which will be processed by the screen.
- c. Definition of the database record which must be accessed and the order of access to achieve the desired processing purpose. Data elements required for currency control are defined and added to the list prepared in (b) above.
- d. Definition of the display screen itself.
- e. Definition of associated screen functions which are to be used to support the primary function. Assignment of function keys and error handling modules.
- f. Preparation of processing logic to achieve the desired results from the IP.

The detail discussion of IP development will follow the above order. As IP parameters are developed, they will be grouped within the parameter deck according to the illustration in Figure A.14.1.

#### DEFINITION OF IP PURPOSE AND USER

Each on-line IP program is described in the data dictionary. This description is tied to the application and service analysis.

1. ADD PROGRAM NAME IS TP. [REQUIRED] This statement defines the IP as a program within the data dictionary. The IP identifier is assigned following the naming convention defined in Appendix B.6.

(8)

ADD PROGRAM NAME IS TPBAAS

2. PREPARED BY. [REQUIRED] The initials of the preparer are entered.

(12)

PREPARED BY 'LET'

3. PROGRAM DESCRIPTION IS. [REQUIRED] This statement is used to define the actual title of the IP screen. This title will be displayed on the first line of the screen. It must not exceed 40 characters.

(12)  
PROGRAM DESCRIPTION IS  
'STORE NEW NIMIS PROJECT'

4. ENTRY-SECURITY IS ENTRY-. [REQUIRED] The security classification of the IP description is entered. See Appendix E.1 for valid security classifications.

(12)  
ENTRY-SECURITY IS ENTRY-UNCLASSIFIED

5. DATA-SECURITY IS DATA-. [REQUIRED] The security classification of the data to be displayed on the screen is defined. See Appendix E.1 for valid security classifications.

(12)  
DATA-SECURITY IS DATA-UNCLASSIFIED

6. LANGUAGE IS COBOL. [REQUIRED] The statement defines the programming language which is used by the generated IP program. See Appendix E.3 for valid languages.

(12)  
LANGUAGE IS COBOL

7. MODE IS SHADOW. [REQUIRED] The use of SHADOW to process the IP is defined.

(12)  
MODE IS SHADOW

8. WITHIN SUBSYSTEM. [REQUIRED] The subsystem who is supported by the IP is defined.

(12)  
WITHIN SUBSYSTEM NIMIS

9. MODULE USED IS TP-. [REQUIRED] This statement defines the module which contains the screen layout. The module name is identical to the program name except that a hyphen is inserted after "TP".

(12)  
MODULE USED IS TP-BAAS

10. SUBSCHEMA IS SS. [REQUIRED] The identifier of the subschema supporting the IP module is added to the "SS" prefix. This subschema is the one primarily supporting the IP and its updating function. It is assigned by the DA staff.

(12)  
SUBSCHEMA IS SSNIM

11. RELATED-IP IS. This statement identifies follow on IP modules which are executed as a result of the user pressing a function key, a PA key, or the result of an error in processing. See Appendix E.17 for a list of valid codes.

(12)  
RELATED-IP IS 'TPBABM,F4'

12. COMMENTS. [REQUIRED] A narrative description of the IP from a user's viewpoint is provided. This description will be used as the basis for the users guide entry on the screen.

A PERIOD MUST FOLLOW THE END OF THE LAST COMMENT LINE.

#### DEFINITION OF IP DATA ELEMENTS

Before actually defining the layout of the display screen it is necessary that the elements to be processed by the screen be identified. This is done by reviewing the intended purpose of the IP and the source document(s) which will be used by the person entering the data. A list of elements is prepared for use in the screen layout step.

A set of parameter cards is prepared for insertion in the IP parameter deck. One parameter card is prepared for

each element to be used by the IP.

(1)  
H, <database element name>

. . .  
 . V .  
 , . F .  
 . N .  
 . .

. . .  
 , .. R .  
 . O .

[,<attribute>] . . .

[,<source book>]

[,LOCK]

[,KEY]

1. H,. This is the identifier for all data element definition parameters. It must begin in position 1 of each such parameter.

2. <database element name>. The name of the database data element to be processed is entered.

3. V,F, or N. This required code identifies:

- a. V- An alphanumeric field whose data values may not fully fill the available data space; e.g., a person's name is not likely to use all 27 characters allowed for such an entry.
- b. F- An alphanumeric field whose data values can be expected to fully fill the available data space; e.g., the code for a state is always two characters in length.
- c. N- A numeric data field.

Program generation software automatically defines the attribute of the data area using this parameter. The

functional processing of alphabetic fields is controlled by the fixed or variable status of the field. An attribute byte is generated after each alphanumeric field. If the field is fixed, the attribute instructs the system to skip to the next unprotected field when the fixed input data has been loaded. If the field is variable, the attribute locks the keyboard if the user over fills the variable field.

4. R or O. This required parameter defines whether the element data value must be present when the complete screen of data is entered. The data is required when the value "R" is present and optional when "O" is present.

5. <attribute>. Attributes other than "UNPROT" are defined. Multiple attributes may be defined, each separated by a period. "UNPROT" is always created. Other valid attributes are:

- a. MDT. The MDT flag is to be set on. This flag forces re-entry of data which has been transmitted to the field from the database.
- b. IC. The cursor is to be set at the beginning of the element, defining the first element on the screen where data can be entered. If omitted, cursor is set at the beginning of the first element on the screen which is not LOCKed (see 7 below).
- c. DRK. The data keyed into the field will not be displayed on the screen. This feature is used when sensitive data, such as passwords or classified information, should not be left on the screen.
- d. BRT. The data keyed into the field will be displayed in high intensity.

6. [, <source book>]. An optional clause which identifies the source library book which is used to verify the value entered through the screen. Appendix E.8 defines the proper source books. If omitted, the data is transferred without verification.

7. [, LOCK]. An optional clause which defines elements used in MODIFY or DELETE IP's as fixed information. Data is





1. A/B. A "A" parameter is always present. This parameter defines the primary path to the target record type. Continuation cards are permitted, each beginning with "A". "B" parameters are present when the database target record requires multiple currencies for a STORE to be executed. The target record must be the last record defined in BOTH "A" and "B" parameters.

2. <record name>. At least one record must be identified for either path. If only one is identified, it is assumed to be the target type. The first record identified in the path must be a BASE (CALC) record.

3. The optional record definition:

- a. <record name>. Definition of the name of record types along the path to the target record. The last record defined is the target record.
- b. (<set name>. The left parenthesis begins immediately after the last character of the record name without an intervening space. The set name is that set which connects the PREVIOUS record in the path to the record associated with the set.
- c. /A-D-N-P. A slash delimits the set name from the definition of the set access mode. Modes are:

(1). "A" defines a set sorted in ascending order. All sets between BASE and INTERMEDIATE or SUBSIDIARY record types (the "A" set) are sorted.

(2). "D" defines a set sorted in descending order. all sets between BASE and INTERMEDIATE or SUBSIDIARY record types (the "A" set) are sorted.

(3). "N" defines a set where the records are stored in NEXT order. All sets between BASE and RELATE record types (the "B" set) are stored either NEXT or PRIOR with a default to NEXT.

(4). "P" defines a set where the records are stored in PRIOR order.

d. <tbl posit>). The literal table position must contain the relative position of the data element which controls this set in the screen definition.

- (1) When the is "A" or "D", the position of the sort key for this set will be found in this position within the screen definitions which follow the "path" parameters.
- (2) When the mode is "N" or "P", the data element to be matched against the database records will be found in this position within the screen definitions.

Once the path to the target record has been established, those data elements which are necessary for currency to locate record occurrences along the path are defined. "H" parameters are prepared for each data element so defined and the KEY option is specified for each. If the IP function is MODIFY or DELETE, these fields are also defined as LOCKED to prevent changing other than the target record.

#### DISPLAY SCREEN DEFINITION

A display screen consists of a CRT display of up to 21 lines of 79 characters. Every line must contain at least one attribute byte, reducing the number of usable characters from 80 to 79.

Certain lines on the screen are reserved:

1. Line 1 is preformatted and contains the IP identifier, the retry count (dark), and the IP title.
2. Lines 23 and 24 are reserved for system messages.
3. Lines 20 through 22 may be used for input or literal information. However, these lines

are used by the IP to return error messages. Any information previously displayed is erased. Should the lines contain data, erroneous information could be passed back to the IP. It is recommended, therefore, that the lines be used only for informational messages.

4. Line 2 may be used if the IP runs short of lines. The display is more readable if this line remains blank.

The display screen should be first defined on a layout sheet of some type. A standard COBOL coding sheet will do the job. These general rules should be applied when entering the data fields on the screen layout:

1. Do not begin any literal or element in position 1. It will be used for an attribute byte by the system.
2. Place all elements which are required to define currency at the top of the screen, making them the first elements to be entered by the user.
3. Assume that an attribute byte will precede all literal values.
4. Assume that an attribute byte will precede all element entry (unprotected) fields. This attribute is defined as the first plus (+) sign of the unprotected field.
5. Each unprotected field is defined by a series of plus signs.
6. Assume each alphanumeric unprotected field will contain an attribute byte at the end of the field. Therefore, the next field on the same line should begin two bytes away (allowing for the terminating attribute of the unprotected field and the beginning attribute of the following field). Future versions of the generator program will remove this inefficiency.

7. Literals may not exceed 50 characters in length.
8. Multiple literals may be defined on a line without intervening elements.
9. The literal immediately preceding an element entry (unprotected) field is assumed to be part of that field.

Three parameter formats are used to define a screen layout:

1. The beginning delimiter
2. The screen line
3. The terminating delimiter.

The beginning delimiter format:

(1)  
C, <screen title>

<screen title> is identical to the contents of the "PROGRAM DESCRIPTION IS" IDD IP definition statement described previously.

The screen line:

(1)  
D<screen description line>

<screen description line> is the 2-79 position illustration of the contents of screen line. The first screen line following the "C" beginning delimiter is assumed to be line 2 of the display.

The terminating delimiter contains "E," in the first three positions of the card. The comma is required.

Once the screen layout is complete, the "H" element definition parameters are ordered so that the first parameter



describes the first element (unprotected) field in the screen display, followed by the second, etc., until all are in order reading from left to right line by line from top to bottom. THIS ORDER IS VERY CRITICAL. MISALIGNMENT WILL HAVE THE DATA LAYOUT SENDING DATA TO THE WRONG FIELDS.

The display definition parameter deck, including the delimiters, will be used as part of an IDD module definition:

1. ADD MODULE NAME IS TP-. The name of the IP is entered.

(8)  
ADD MODULE NAME IS TP-BAAS

2. PREPARED BY. [REQUIRED] The initials of the preparer are entered.

(12)  
PREPARED BY 'LET'

3. MODULE DESCRIPTION IS. The title of the module is entered.

(12)  
MODULE DESCRIPTION IS  
'STORE NEW NIMIS PROJECT'

4. MODULE SOURCE FOLLOWS. The statement is the delimiter for the screen display parameters prepared above. They are placed immediately following this statement.

5. MSEND. The statement is the terminating delimiter for the screen display parameters prepared above. A PERIOD MUST FOLLOW "MSEND".

#### DEFINITION OF ASSOCIATED FUNCTIONS

Frequently it is desirable to permit the user to perform additional input processing which is directly related. For example, after storing a work project it is likely that the user will also want to define the persons assigned to the project, describe the project more

completely, and identify any references which are associated with the project.

Each of these functions may be requested by the user by assigning them to individual functions keys. The functions can also be separately executed when desired.

Each associated functions should be defined. Then function keys, assigned in accordance with the convention defined in Appendix B.11, are identified. Parameters are prepared:

```
(1)
M,      <function code>
        ,<associated transaction>
        [, <literal>]
```

1. M, the parameter identifier must be terminated with a comma.
2. <function code> is one of the codes defined in Appendix E.17.
3. <associated transaction> is a 4-character identifier of another IP or other program which is to be executed if the defined function option is selected.
4. <literal> is an optional 36-character message which will accompany the function key identifier on the screen.

When the function code is F4-9, FA-C, A1-2, or EN, the code and literal are displayed on the screen one line following the last detail line of the IP. Two function messages are placed on one screen line.

If no entry is made for the "enter" key (EN), a default assumes that pressing the enter key will repeat the same transaction function if no disabling errors have occurred.

Functions keys 1 through 3 are reserved for system use. See Appendix B.11.

## PREDEFINED SYSTEM FUNCTIONS

The IP generator program performs a number of functions for the programmer, eliminating the need for insertion of all but very unusual coding logic. Functions performed are:

1. The database identifier (DBID-nnnn) is loaded based upon the value provided by the user at sign on.
2. The current system date is loaded to DATE-STORED-nnnn when a record is stored in the database.
3. The current system date is loaded to DATE-MOD-nnnn when a record is stored or modified.
4. The logical record type (LOG-TYPE-nnnn) is loaded when the record is stored.
5. The database control record is updated:
  - a. The system date is loaded to the date of latest modification.
  - b. The update activity count is incremented by one.
  - c. The number of records is incremented when a record is stored and decremented when a record is modified.
6. All data elements are validated according to the instructions in the "H" parameter.
7. Data from the input area is transferred to database record buffers for activity at the proper time.
8. Currency is established to properly process the target record.
9. Database maintenance DML commands are executed at the proper time.

10. Selection of follow on transactions (programs) is made according to "M" parameter definitions.

#### IP SPECIAL PROCESSING LOGIC

Special processing logic may be inserted into the generated IP program. This logic is inserted at five locations within the program:

1. "N" parameters are used to define additional data work fields. These fields are NOT initialized and may not contain VALUE clauses. It is recommended that any fields defined by this parameter be initialized in the "T" parameter group below. All DATA DIVISION parameters are coded as COBOL 03 level data statements.

NOTE: The aggregate number of bytes of all "N" parameters may not exceed 30 bytes.

The following three groups of parameters are coded as COBOL PROCEDURE DIVISION statement following standard COBOL syntax. The statements are moved intact exactly as written.

2. "R" parameters are inserted into the generated program immediately before the DML statement updating the database is executed. The insertion follows the automatic transfer of data from input to database areas.
3. "S" parameters are inserted immediately following the DML statement updating the database and its error checking logic.
4. "T" parameters are inserted immediately preceding the establishment of currency with the database records upon which the target record is dependent. The parameters are inserted after the automatic transfer of key data from the input to database record key areas.
5. "U" parameters are inserted at the end of the

generated program. "U" parameters are intended to be complete SECTIONS of procedure code which is PERFORMed by statements in the "R", "S", or "T" groups above. No direct execution path is provided to code in this group from the generated program logic.

The parameter deck is terminated by a "Z" parameter.

NOTE: All data element validation and movement from input areas to the database records is performed automatically based on the processing option (STORE, MODIFY, or DELETE) and the type of element validation requested in the "H" parameter.

#### PARAMETER DECK ASSEMBLY

The IP parameter deck is assembled as illustrated below in Figure A.14.1.

```
A,NIM-PROJECT
B,ORGANIZATION,NIM-PROJECT(NIMIS-1000-1/A)
C,STORE NEW NIMIS PROJECT
DPROJECT IDENTIFIER-+++++ REQUESTER-+++++
DDATE ORIGINATED-+++++
E,
H,NIM-PROJ-ID,V,R,UNPROT.IC,SSZA,KEY
H,NAME-ORGAN,V,R,UNPROT,SSZA,LOCK,KEY
H,DATE-ORIG,N,R,UNPROT,SSZF
M,F4,RABS,STORE NIMIS COMMENTS
M,F5,BABM,ADD DATES
M,EN,BAAS,STORE ANOTHER PROJECT
N      03 WORK-AREA      PIC XX.
R      MOVE A TO B.
S      MOVE B TO D.
S      PERFORM DO-LOOP.
T      MOVE D TO E.
U      DO-LOOP SECTION.
U      MOVE WORK-AREA TO F.
U      XDL.  EXIT.
Z,
```

FIGURE A.14.1



## APPENDIX A.17

### USERS GUIDE PREPARATION INSTRUCTIONS

Users of integrated database systems may access the database in either batch or on-line modes. Frequently the same user may utilize both methods under different conditions. It is important, therefore, that user instructions for utilizing the integrated database be complete, accurate, and easily understood by non-ADP personnel.

Each users guide developed for integrated database applications will follow the consistent format and structure guidelines described in this Appendix. Appendix F provides an example of users guides in use and the format being used.

#### Structure of the Users Guide

Each users guide will consist of a body of six main sections plus four appendixes. The body of the guide contains:

1. Introduction. The introduction contains a brief description of the database supported by the guide, the scope of the guide, and its intended audience.
2. Capability Description. This section describes briefly, in overview form, the capabilities of the integrated database application(s) supported by the guide. The emphasis here is on brevity and clarity aimed at the non-ADP user.
3. Batch Input Processing. This section provides an overview, including a brief glossary of commonly used terms, of the batch input processing (IP) functions. Following the introductory narrative, each input format used by the application(s) is individually described. The format of this description will be discussed in detail under "Batch Input Processing Instructions".
4. Batch Report Preparation. This section provides an overview of batch report

preparation capabilities. It then provides a detail description of each standard report included as part of the application system. The format of this description will be discussed in detail under "Batch Report Instructions".

5. On-line Input Processing. This section is structured in the same manner as that for batch input processing. Pictorials of on-line screens are described as defined in "On-line Input Processing Instructions".
6. On-line Report Processing. This section is structured in the same manner as that for batch reporting. It defines, initially, how to display reports placed in the report display queue. This is followed by directions for requesting reports and creating new reports. The corresponding section in Appendix F may be used with minor changes in wording for the narrative portion of this section. Further details are provided in "On-line Report Processing Instructions".

The appendixes to the users guide contain:

1. Appendix A. This appendix contains clean copies of all forms described by the users guide.
2. Appendix B. This appendix contains instructions for preparation of batch job control decks necessary to process those functions in the users guide which are batch oriented.
3. Appendix C. This appendix contains a summary of standard and special messages produced by the integrated database system and instructions for responding to the messages.
4. Appendix D. This appendix presents a summary of ad hoc on-line query facilities and techniques.

## Preparation Format of the Users Guide

Each users guide contains the following pages or sections in the order listed:

1. Cover page, containing the name of the application systems(s) supported by the guide, version of the guide and its date (month and year) of publication, and the identification number of the guide manual.
2. Record of changes, initially blank. As updates to the guide take place, each is entered on this change page.
3. Foreword, a brief statement of the manual approach which uses the wording of the Appendix F sample except for tailoring to meet the application(s) covered by the guide.

NOTE: The above pages are prepared vertically on 8 x 10 1/2 paper, printed on the front side only. Those items below are printed horizontally, using both sides of the paper, as appropriate. See "Page Printing Organization".

4. Table of contents, identifying each section of the users guide. Subsection definitions are provided to the individual function description level.
5. Sections 1 through 6.
6. Appendixes A through D.

## Page Printing Organization

The main body of the users guide is printed horizontally on 8 x 10 1/2 paper. Narrative at the beginning of sections and appendixes is displayed in two-column format.

When describing the horizontal format, the terms "left hand" and "right hand" are not effective to describe page positioning. Instead, the page positioning will be described as "top page" and "bottom page". Taking a standard three-ring binder and turning it 90 degrees so that what was the left hand page is now at the top, the binder is properly

aligned. The left page now becomes the top page and the right page becomes the bottom.

All sections and new subjects start with the bottom page. Where necessary, the preceding top page will be left blank.

An one-inch margin will be left for each page (top of the bottom page and bottom of the top page) for binding. A 3/4-inch left and right margin will be allowed.

The title of the section will be placed one-half inch from the bottom of the bottom page, right justified to the margin. Individual instruction groups will identify the group (see individual group instructions) instead of the section title.

The page number of the page will be centered on the same line as the title of the section at the bottom of the bottom page. On the top page, the page number will be centered one-half inch from the top of the page.

The section "number" and title will be centered on the bottom page two inches from the top of the page on the first page of the section only.

#### Batch Input Processing Instructions

Each batch input processing instruction is composed of three items:

1. A narrative description of the function(s) supported by the specific input processing format. This narrative should be extracted from the comments section prepared during IP definition (Appendix A.12) and stored in the IDD. The narrative should clearly describe the purpose of the format in non-ADP terms. The narrative is written in single-column style and centered on the bottom page.
2. A pictorial of the IP format sheet. This pictorial is an exact duplicate of the blank format sheet found in Appendix A. The sheet is completed with representative information. Each field on the sheet is referenced by a circle, containing a reference number, and an

arrow pointing to the field in such a way that no confusion exists about which field is being described. The pictorial is placed on the top page.

3. A detailed narrative description of user instructions for completing each field in the pictorial. The narrative description is keyed to the numbered arrows on the illustration. The narrative is presented in single-column format. Field key numbers are placed at the left margin. The field name AS SHOWN ON THE ILLUSTRATED FORMAT is shown, followed by the narrative description. Where possible, the complete information required to complete the field entry is provided. When this information includes a long or detailed table, the table may be placed in an appendix (after Appendix D) and referenced in the narrative.

When the number of fields and/or the narrative exceeds a single page, the pictorial is duplicated at the top of the next page and the narrative continued on the next bottom page.

The batch input processing instructions are grouped by the record types, within the database, that they support. All formats supporting a specific record type should be grouped together. This means that store, modify, and delete IP formats for the same record type will be adjacent. Store formats are first, followed by one or more modify formats, and ended by the delete format, if applicable.

Each major group of IP formats should be preceded by an index page with a visible tab for rapid access.

Each users guide will contain a visible tabbed place at the beginning of the batch IP section where the user may place frequently used IP format instructions. This will make locating such instructions easier for the user.

As part of the introductory narrative for each IP format, precautionary notes should be included where the action performed by the format could possibly damage the database. This is particularly important for delete IP's.



Such a warning message will be preceded by "CAUTION:" and placed as a separate paragraph at the end of the narrative.

#### Batch Input Processing Coding Formats

Batch input processing coding formats are developed using a grid template on 11 x 14-inch unlined paper. When the format has been completed, it is reduced to 8 x 10 1/2-inches.

The grid template provides a series of horizontal lines 3/16-inch apart intersected by a series of vertical lines 1/10-inch apart. This grid pattern creates individual blocks for coding which are large enough for easy fill-in while allowing a maximum number of potential blocks per page and line. The 10 blocks per inch corresponds to a 10-pitch typewriter and facilitates use of an "ORATOR" type font for pre-defined code within blocks.

The normal approach followed when developing a batch IP format is to place elements which are common to all following lines at the top of the format and indented five blocks. The comment "Reproduced in all data lines below" is enclosed in brackets beside the last element on this line.

The remaining elements in the IP line are aligned horizontally across the page on one or more coding lines. The IP identifier is placed at the left margin. Individual elements are placed on the line in ascending position order. If the data line requirements exceed the format line space, use the following format line and indent the beginning of the data element to align with the first element on the preceding line. Elements must be separated by at least one block space on the format line. Where practical, two or three spaces is desirable. In some cases where elements lengths are short and their corresponding titles are long, extra separation to improve readability is desirable.

The descriptive title of each element is placed above the fill-in blocks. It is centered and, desirably, is typed using 12-pitch Prestige Elite font.

The corresponding card positions of the beginning and end of each element are placed below the fill-in blocks. 12-pitch Prestige Elite font is desirable.

The first line of fill-in blocks should begin 1 1/2 inches from the top margin and end 1 1/2 inches from the bottom margin. If additional lines are required, they must be taken from the top margin first. Since the bottom margin will be punched in the examples, the bottom margin must not be reduced to less than 1-inch.

The functional purpose of the IP format is centered 1 inch from the top of the page. The Orator font at 10-pitch is used for this line.

At least one empty line block should be left between the format data lines to leave room for titles and card positions. When multiple lines of the same data elements in the same card positions are used by the IP, successive line blocks may be used to conserve space and improve readability of the format. Corresponding positions number references are placed below the last line of the multiple line group.

#### Batch Report Instructions

Each batch report instruction is composed of five items:

1. A narrative description of the report and the function(s) it supports. This narrative should be extracted from the comments sections prepared during report definition (Appendixes A.10 and A.11) and stored in the IDD. The narrative should clearly describe the purpose of the report in non-ADP terms. The narrative is written in single-column style and centered on the bottom page.
2. A pictorial of the report page. This pictorial is an exact duplicate of an actual page of a representative execution of the report. The example must illustrate all normal and optional print lines and features. Each field on the report is referenced by a circle, containing a reference number, and an arrow pointing to the field in such a way that no confusion exists about which field is being described. The pictorial is placed on the top page.

3. A detailed narrative description of each report field referenced in the report page pictorial. The narrative is presented in single-column format. Field key numbers are placed at the left margin.
4. A pictorial of the control cards required to produce the report. This pictorial will illustrate each separate control card in the order they appear in the deck. See Appendix B.2 of Appendix F. The pictorial is placed on the top page. Each variable field within the control card pictorial will be identified by a reference number, using the circle and arrow technique used in 3 above.
5. A detailed narrative of the control card pictorial with reference numbers at the left margin. Each variable field must be thoroughly explained.

NOTE: Reports typically exist in "families" which utilize the same format but produce different results based on sorting and logical data selection criteria. Where a family of reports exists:

1. A index tab page will be inserted, keyed to the report family. The tab page will be a bottom page and contain a table of contents of the reports in the family with sufficient description for the user to select the desired report.
2. Each individual report option within the family will be defined and illustrated as a separate entity.

#### On-line Input Processing Instructions

On-line input processing differs from batch primarily in the volume of information which may be stored in the database from a single IP. It is also possible to direct the system to proceed from one IP to another through use of function keys.

Each on-line input processing instruction is composed of four items:

1. A narrative description of the function(s) supported by the specific ip screen. This narrative should be extracted from the comments section prepared during IP definition (Appendix A.14) and stored in the IDD. The narrative must clearly describe the purpose(s) of the screen in non-ADP terms. The narrative is written in single-column style and centered on the bottom page.
2. A pictorial of the IP screen. This pictorial is an exact duplicate of what is displayed on the screen. Representative information is included for illustrative purposes. Each field on the screen is referenced by the number, circle, and arrow technique described previously. The pictorial is placed on the top page. The screen pictorial is surrounded by a CRT screen border (see Section 6 of Appendix F).
3. A two-level chart showing the primary screen IP and each of the functional options provided by function keys. This chart will be no larger than 2 1/2 inches high by the width of the paper. It will be placed at the top of each bottom page which contains screen instructions.
4. A detailed narrative description of user instructions for completion of each field in the pictorial. The narrative description is keyed to the numbered arrows on the illustration. The narrative is present in single-column format. Field key numbers are placed at the left margin. The field name AS SHOWN ON THE ILLUSTRATED FORMAT is shown, followed by the narrative description. Where possible, the complete information required to complete the field entry is provided. When this information includes a long or detailed table, the table may be placed in an appendix (after Appendix D) and referenced in the narrative.

When the number of fields and/or the narrative exceeds a single page, the pictorial is duplicated at the top of the next page and the narrative continued on the next bottom page.

On-line IP instructions are grouped by the functional options which they present. An IP screen which permits the calling of several subordinate screens will precede those subordinate screens in the users guide. Under some circumstances, it is possible for a single screen's instructions to be required in several places for continuity. When this occurs, the IP instructions will be duplicated and placed in the appropriate locations.

Each major group of IP screens should be preceded by an index page with a visible tab for rapid access.

Each users guide will contain a visible tabbed place at the beginning of the on-line IP section where the user may place frequently used IP screen instructions.



## APPENDIX B.1

### DATA ELEMENT NAME CONVENTIONS, GENERAL

It is important to consistency within the structure and representation of the contents of the database that data elements be named and addressed in a uniform manner. The data element naming conventions described here are established to meet this basic requirement. Where an unusual condition makes naming a data element without regard for these conventions, written approval of the Database Administrator must be obtained.

#### Conventions:

1. The data element name will reflect, as clearly as possible, the purpose or contents of the data element.
2. The data element name will not exceed 16 characters in length, including hyphens.
3. Where the element is one of a common type listed in Appendix B.2, the appropriate prefix will be used.
4. All data element names must begin with an alphabetic character,

## APPENDIX B.2

### COMMON DATA ELEMENT NAME PREFIXES

Certain data elements within the database are basically identical to other elements except for their individual usage. For example: "NAME" may be the name of a person, the name of a publication, the name of a company, or the name of a position, etc. All, however, are NAMES. The same is true of "DATE", even though the type of date may vary widely.

These common data types are identified by a one-to-four character prefix which is appended to the data element or element group name to identify its generic group. When used as part of an element name, the prefix is separated from the main portion of the name by a hyphen (-).

The prefixes listed in this appendix are broken into two groups: primary and secondary. Primary prefixes are used wherever applicable. Secondary prefixes are used if a primary prefix is not also used. The secondary prefix is optional when a primary prefix is used. Where both primary and secondary prefixes are used in the same data name, the hyphen between the prefixes may be omitted.

#### Primary prefixes are:

1. DATE. This prefix is used for all data element groups defining dates. The form of date: calendar; computed; or Julian, is optionally indicated as J (Julian) C (computed), or G (Georgian calendar) as an additional prefix character.
2. MO. Month of the year.
3. DY. Day of the month.
4. YR. Year of the century.
5. NAME. This prefix is used for all data elements of the generic group defined as names.

6. ADDR. Addresses.
7. CITY. A city, town, borough.
8. STAT. A state or province.
9. CTRY. Country.
10. CTY. County.
11. DATA. Data identifier.
12. DBKY. Database key information.
13. DTG. Date-time-group element group.
14. TIME. Clock time of day element group.
15. HR. Hour of the day.
16. MIN. Minutes of an hour.
17. SEC. Seconds within a minute.
18. ZIP. Postal zip code.

Secondary prefixes are:

1. SW. A switch or flag element.
2. CTR. A counter or accumulated data element.

## APPENDIX B.3

### DATA ELEMENT PICTURE DEFINITION CONVENTIONS

The size and composition of data elements are commonly defined through a short-hand form called a "picture". This short-hand form may be directly translated into computer-readable format. Picture definitions are used within the data dictionary to define the structure of the data element within the database.

The picture definition is composed of three basic element segments:

1. A qualifier,
2. The data type,
3. The data length.

The qualifier which may be used applies to numeric type data only. The letter "S" may precede the data type to identify a numeric data type as signed. The use of a signed numeric data type permits the data element to contain negative values as well as positive values. When used, the qualifier, is immediately followed by the numeric data type without intervening spaces or special characters.

Three basic data types are permitted:

1. Numeric. All data within the element will always be numeric (zero through nine). Alphabetic or special characters entered in this field will result in processing errors. The picture symbol for a numeric data type is a nine (9).
2. Alphabetic. All data within the element will always be alphabetic (the alphabet in upper-case and normal punctuation characters). Numeric data entered in this field will result in processing errors. The picture symbol for an alphabetic data type is the letter "A". This option is seldom used except where numeric data is to be specifically excluded.

3. Alphanumeric. Data within the element may consist of any of the permissible characters or digits defined by the computer. Alphabetic and numeric data may be intermixed at will without processing errors. Numeric data stored in this type of field should be used for information purposes only. Computation processing of numeric data in this data type will require extra handling by the computer and will result in slower processing. The picture symbol for an alphanumeric data type is the letter "X". This is the most commonly used data type.

The data element length is defined following the data type segment. The length is subject to certain rules:

1. Numeric data elements may not exceed 18 digits in length.
2. Alphabetic or alphanumeric data elements may not exceed 800 characters in length without approval of the project manager.

Data element length is identified by a numeric value enclosed in parenthesis. This value may range from one through the limits defined above, e.g., (12), (8). The size segment immediately follows the data type segment without intervening spaces or special characters.

The complete picture definition for the various types of data are illustrated below:

1. Numeric. S9(8) 9(8)
2. Alphabetic. A(10)
3. Alphanumeric. X(10)



## APPENDIX B.4

### DATA ELEMENT USAGE CONVENTIONS

The usage clause defines the intended use and composition of a data element within the database. The usage clause is a qualifier for the PICTURE clause.

Available usage clauses are:

1. DISPLAY. This clause is the default whenever the usage clause is omitted. If this usage is desired, omit the clause.
2. COMP. This clause is used to define numeric data types which will use a special form of numeric representation within the database. This usage type should be used only with the approval of the project manager.
3. CONDITION-NAME. It is sometimes useful to define a phrase which identifies a condition or value of the data element. This condition is set whenever the data element contains the value within the database record matches the value associated with the condition name. The project manager should be consulted prior to defining any condition names.

## APPENDIX B.5

### INITIAL VALUE DEFINITION

It is desirable that an initial value be provided for all data elements. This insures that garbage data values will not be loaded to the database. Initial values are typically either SPACES (blanks) for alphabetic or alphanumeric data elements or ZERO (0) for numeric data elements. If the numeric item is signed (a "S" in the qualifier segment), the initial value should be "+0".

The length of the value data must not exceed the length of the field as defined in the PICTURE clause. Therefore, if the field is ten characters long, the value literal must not exceed ten characters. If an alphabetic or alphanumeric literal contains spaces or special characters (characters other than A-Z, 0-9, and hyphen, period, or comma), it must be enclosed in single quote marks.

The literal may not exceed 34 characters in any event.

## APPENDIX B.6

### IP NAMING CONVENTIONS

Each input processing (IP) module is identified as a unique entity through a defined name. This name is composed of four characters.

The three primary characters are assigned by the DA staff. They are normally alphabetic. The third character may be numeric if approved by the DBA.

The first two characters must always be alphabetic and will consist of the logical record type identifier for the database record supported by the IP.

The third character identifies individual IP modules supporting the specified database record type. The alphabet is used, A through Z, followed by the numbers 0 through 9 if required. The "A" ALWAYS identifies the IP which primarily STORES the record occurrence in the database. Where more than one IP stores the record, the identifier used should alphabetically precede other IP's which modify the record in the same context.

The fourth character identifies the processing option to be performed by the IP. This character may be:

1. S - Store a record in the database.
2. M - Modify an existing record in the database.
3. D - Delete an existing record from the database.
4. Q - Retrieve a record from the database.

## APPENDIX B.7

### COBOL PROGRAM DEVELOPMENT CONVENTIONS

Program development must consider future maintenance requirements as well as today's performance. A major factor in the resource requirements for maintenance is the readability and understandability of source programs. This Appendix defines programming conventions which are designed to enhance the effective design of a program while simultaneously making it self-documenting.

It is the intent of these conventions to document a COBOL program sufficiently that separate program maintenance documentation is not required. Experience has shown that separate documents are not kept current and suffer from a lack of detail. Inserting the documentation into the source program itself eliminates that problem and reduces the overall documentation effort.

COBOL programs developed for execution on NIPSSA computers will follow specific coding and documentation conventions. These conventions are defined by major COBOL division:

#### 1. IDENTIFICATION DIVISION.

IDENTIFICATION DIVISION.  
PROGRAM-ID. assigned name.  
INSTALLATION. NIPSSA.  
DATE-COMPILED.  
SECURITY. classification of source program.  
AUTHOR. name of originating author.  
REMARKS.

program description using \* comment statements  
description will define purpose of program,  
source of data, supporting files.

a version/revision log identifies the latest  
revision level, the date, and the purpose of  
the revision. place a blank line between  
revision description.

## 2. DATA DIVISION.

### WORKING-STORAGE SECTION.

#### 01 V-L-IDENTIFICATION.

03 FILLER	PIC X(8)	VALUE 'program name'.
03 FILLER	PIC X(16)	VALUE 'VERSION 01'.
03 FILLER	PIC X(16)	VALUE 'LEVEL 01'.
03 FILLER	PIC X(16)	VALUE 'DATE- mm/dd/yy'.

#### 01 SWITCHES.

03 XYZ-SW	PIC 9	VALUE 0.
88 XYZ-OFF		VALUE 0.
88 XYZ-ON		VALUE 1.

#### 01 COUNTERS COMP-3.

03 ABC-CT	PIC S9(5)	VALUE +0.
03 MAIN-CTR	PIC S9(8)	VALUE +6.

#### 01 SUBSCRIPTS COMP.

03 R	PIC S99	VALUE +1.
------	---------	-----------

#### 01 PROGRAM-MESSAGES.

03 MSG1	PIC X(16)	VALUE 'TOO MANY ERRORS'.
---------	-----------	--------------------------

#### 01 PROGRAM-TABLES.

#### 01 HOLD-AREAS.

03 MAIN-HOLD	PIC X(8)	VALUE SPACES.
03 DATA-HLD	PIC X(80)	VALUE SPACES.

## 3. PROCEDURE DIVISION.

Procedure code will utilize structured programming techniques whenever practical. Deviations from structured approaches will be thoroughly commented in the program and approved by the application project manager.

These conventions will be followed:

1. Where practical, the use of "GO TO" will be avoided. Backward "GO TO" statements (where the object of the GO TO precedes the GO TO statement itself) will not be used.
2. PERFORM verbs will address only SECTIONS. Performing of paragraphs or use of "THRU"



option will be avoided.

3. Section names will be suffixed by a numeric value which acts as a running index for reference purposes. The numeric value will increase from front to back of the procedure division. Example: MAIN-PROCESS-100 SECTION precedes READ-FILE-240 SECTION.
4. Each section will begin a new page. The logic code of the section will be preceded by a comment narrative describing the function(s) to be performed by the section. This description will include the input and output of the section, where applicable, and the calling and called sections.
5. Within the logic of a section, each related group of source statements will be preceded by a block of comments describing the groups function(s). The comment block will be preceded and followed by a line of hyphens to set off the comments. Comments will begin in position 20 of the comment line.
6. Section names will be descriptive but limited in length to 16 characters plus the numeric suffix.
7. COBOL source statements which are not dependent on other statements will begin in position 12. Statements which are dependent upon a preceding statement will be indented 4 positions.
8. ONLY ONE STATEMENT MAY BE CODED PER SOURCE LINE.
9. When ELSE statements accompany IF statements, the ELSE will be vertically aligned with its corresponding IF to insure understanding.
10. Use of SKIP statements to visually improve the readability or understanding of the source program is encouraged.

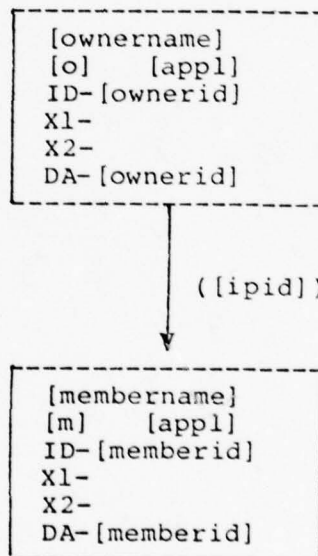
11. Positions 1-5 of each source line will contain the version and level number of the line. This is particularly important during revision and enhancement changes to the program.
12. Program section organization will follow this general order:
  - (a) Main processing path. A short section composed of PERFORMs of a series of subordinate sections.
  - (b) Housekeeping sections. Each program, as required, may have housekeeping functions to be performed at the start and finish of the program execution. These include data initializing, file opening and closing, etc.
  - (c) I/O sections. All program I/O, other than IDMS DML statements, should be performed in individual subprograms to facilitate future changes in I/O interfaces. For example, card reading should be performed by a single routine as should tape reading or writing.
  - (d) Major processing functions. Each major processing function should be inserted in the same order as it is performed in the main processing path. Sub-functions should immediately follow its associated main function IF it is used only by that function. It is recommended that each main function be assigned a numeric suffix such as 1000, 2000, etc. This leaves the intervening numbers for use by sub-functions (1100, 1200). Next level subfunctions (1110, 1120) follow.

(e) Commonly used subfunctions. These subfunctions are used by more than one of the major functions. These should use a numeric suffix which begins with nine, e.g., 9000 or 90000 (if needed).

APPENDIX B.8

SCHEMA RECORD STRUCTURE TEMPLATE

APPLICATION- [appl]  
OWNER- [ownername]  
MEMBER- [membername]  
DATE- [date]  
BY- [person]



Processing is [mode] thru [ipid] IP

PURPOSE:

## APPENDIX B.09

## GENERAL UDB RECORD TYPES

During design of the database for different applications, it became evident that certain data entities were frequently repeated between applications. The data itself is frequently unique but the function supported remains the same.

Because of the flexibility of UDB to link any record type to any other record type, it is possible to define a series of common record entities which can then be used by many applications without additional design effort. The same input processing software may be used by a majority of the applications, reducing software development costs in many cases.

## Comments Record

A general comments record has been defined. This record type is identified as logical record type "RA". Other comment records will be defined as logical types "RB through "RZ" and "R0" through "R9". All IP's which process a comments record occurrence will begin with "R".

The general description of the comments record below may be modified where necessary to accommodate additional elements. The 800-byte general text area should NOT be altered. The FI-xxxx area will ALWAYS be the first in the record description and will not be altered. The first three fields in the ID-xxxx block will always be in the same position and of the same length.

The UDB-COMMENTS record consists of:

1. FI-xxxx. The fixed data block contains three fields:
  - a. CLASS-nnnn, the security classification of the record occurrence, the same as SECURITY-CLASS. The field is stored by the IP. If no classification is defined, the default is unclassified.



- b. HANDL-nnnn, the releasability code of the record occurrence, the same as RELEASABILITY. The field is stored by the IP. The default is no restriction, spaces, if no releasability data is provided.
- c. DATE-MOD-nnnn, the date on which the record occurrence was last modified, in format yymmdd, the same as DATE-COMMON.
- d. DBID-nnnn, the 8-byte database identifier which links the record occurrence to a specific database.
- e. LOG-TYPE-nnnn, a 2-byte logical type code for the comment record. The field may be user specified where different kinds of comments may be applied to a specific owner record. The codes should be alphabetic. If no code is used, the logical record type code "Rx" should be used.
- f. DATE-STORED-nnnn, the date on which the record was stored in the database, format yymmdd, same as DATE-COMMON. The field is automatically stored by the store IP and may not be modified by the user.

2. ID-xxxx. The identifier block contains:

- a. DATE-COMMENT-nnnn, a 6-byte field, same as DATE-COMMON, which contains the date of the comment/description. This field may be omitted where the date is not a factor in establishing a unique record occurrence.
- b. SEQUENCE-nnnn, a 3-byte sequence field, stored by the user IP, to distinguish comment records which

require multiple occurrences to completely enter a comment. The data is numeric. The first occurrence is stored

3. DA-xxxx. The text block, 800 bytes long, may be loaded in any form by the user. It is not recommended that multiple data fields be placed in this block unless the block itself is redefined in the schema. The schema looks at this block in one of two ways: (a) a single contiguous block of 800 bytes; (2) 20 40-character lines of data.

Comments records will be stored as automatic members of another record entity. Because of this requirement, individual IP's will be created.

xxxx = the name of the record type

nnnn = the record identifier of the record type.

The format of the coding sheet, and therefore of the IP input definition, will be largely the same from one IP to another.

Coding sheets will be basically structured as follows:

1. Positions 1-4 will contain RxxS/M/D, where xx is the unique identifier for the IP.
2. Positions 5-25 will contain the identifier of the owner occurrence to which the comment will be attached. When less than 21 characters are required, show only the characters (divided into logical element fields if appropriate) which are necessary to uniquely identify the owner.
3. Positions 26-31 will contain the date on which the comment is stored (created), in format yymmdd.

4. Positions 32-33 will contain the logical comment subtype. This is used if the application relationship requires that multiple kinds of comments be identified, such as missions, functions, goals, etc. This field is used only when a comment record is used for multiple purposes for a specific owner. When used, the subtype is placed as the first element in the ID-xxxx data identifier group.
5. Positions 34-36 will contain the sequence of occurrences of the comment itself. When the narrative exceeds 800 bytes, this field permits additional uniquely identified comment record occurrences. The sequence number begins with "000" and is incremented for each additional entry within the same comment.
6. Position 37 will contain the security classification of the comment. If omitted, this defaults to unclassified. The actual entry is necessary only on the line 01 (see 7 below) card.
7. Position 38 will contain the security releasability code for the comment. If omitted, no releasability restrictions are applied.
8. Positions 39-40 contain the logical line number of the comment. Each comment may contain up to 20 40-character logical lines. The store function card MUST be associated with line 01. Modifications to add more lines (or change line 01) will contain the logical line number to be modified.
8. Positions 41-80 contain the logical line text. The data is free-form and can contain any valid alphabetic, numeric, or special characters within the EBCDIC set. While the logical line limit is 40 characters, this does not mean that the user's textual lines must conform to that limit. The line length on printed (or other) output may be whatever

length desired. Keep in mind that user line lengths other than multiples of 40 bytes will have to be addressed through subscripts and character- by-character processing rather than in 40-character groups.

#### Reference Record

A general record type has been defined to support all types of documents, articles, letters, memos, invoices, etc. A list of the specific reference subtypes is provided at the end of this section.

The reference record is stored through a group of predefined IP's. The IP's may be used without change by all applications. Alternatively, additional tailored IP's may be written to process those data elements which relate to a specific application and ignore those which do not apply.

The reference record is stored as an independent occurrence without linkage to other entity occurrences. This permits the independent association of multiple relationships with a single reference by other applications. For example: A letter received through the mail is entered into the database as a reference. It appears as part of a correspondence control function. When routed to someone for action, it is linked to the department and/or individual who will be responsible for the letter. When a reply is prepared, the letter will become a reference to the reply.

When a reference is specifically created as part of an application function (and is therefore DEPENDENT upon that function for its existence), a specific IP may be written. The IP will have the form QxxS/M/D, where xx is the IP's unique identifier.

The reference record contains the following data elements:

1. FI-REFERENCE. The fixed group is composed of 6 elements:
  - a. CLASS-1002. The security classification of the record occurrence, same as SECURITY-CLASS. The field value is stored by the IP. If no value is entered,

a default of unclassified is assumed.

- b. HANDL-1002. The releasability code of the record occurrence, same as RELEASABILITY. The field value is stored by the IP. If no value is entered, a default of no restriction (blank) is assumed.
  - c. DATE-MOD-1002. The date on which the record occurrence was last modified, in format yymmdd, same as DATE-COMMON. The field is stored automatically by the system.
  - d. DBID-1002, the 8-byte database identifier. The field is stored by the system.
  - e. LOG-TYPE-1002, the logical type for the reference, stored by the IP, based on the table of types following this description.
  - f. DATE-STORED-1002, the date on which the record was stored in the database, in format yymmdd, same as DATE-COMMON. The field is stored by the system.
2. ID-REFERENCE, the identifier which uniquely identifies one reference occurrence from another:
- a. REF-SERIAL, a 15-byte serial identifier assigned by the originator.
  - b. A 17-byte field available for further identification.
2. DA-REFERENCE, the main detail data description:
- a. NAME-REF-SIGN, a 27-byte field



containing the name of the person signing the reference

- b. REF-CLASS, a 1-byte field identifying the security classification assigned to the reference itself. This is different from the security classification assigned to the reference entity stored in the database.
- c. REF-HANDL, a 1-byte field identifying the security releasability assigned to the reference itself. This is different from the security releasability assigned to the reference entity stored in the database.
- d. REF-SUBJ, a 40-byte field identifying the subject of the reference.
- e. REF-SUBJ-CODE, a 4-byte field identifying the standard subject code for the reference.
- f. REF-SUBJ-CLASS, a 1-byte field identifying the security classification of the subject line itself.
- g. REF-SUBJ-HANDL, a 1-byte field identifying the security releasability code for the subject line itself.
- h. DATE-SUB-REC, a 6-byte date field identifying the date on which the reference was received, in format yymmdd.
- i. DATE-OF-REF, a 6-byte date field identifying the date of the reference itself, in format yymmdd.

- B9.1
- j. DATE-REF-FORW, a 6-byte date field identifying the on which the reference was forwarded to another organization or was disposed of, in format yymmdd.
  - k. REF-TYPE, a 2-byte field identifying the type of material which is being placed in the database as a reference.

#### Reference types

LETTER	AA	MESSAGE	BA
MEMORANDUM	CA	PDM	CB
MFR	CC	ROUTE SLIP	DA
INVOICE	EA	REGULATION	FA
INSTRUCTION	GA	MANUAL	HA
PAMPHLET	IA	BOOK	JA
PERIODICAL	KA	ARTICLE	LA
PLAN	MA	REPORT	NA
SCHEDULE	OA	TABLE	PA
LIST	QA	PROGRAM	RA
CHART	SA	DRAWING	SB
MAP	SC	PHOTO	SD
ENCYCLOPEDIA	TA	CATALOG	TB
GLOSSARY	TC	INDEX	TD
APPENDIX	TE		

- l. REF-UNIT-COST, a 6-byte numeric field containing the dollars and cents cost of acquisition of the reference. This field is used primarily to account for books, etc., which have been purchased.
- m. REF-COPIES, a 3-byte numeric field defining the number of copies of the reference which are available.
- n. REF-ENCLS, a 2-byte numeric field defining the number of enclosures which are part of the reference.

- o. DATE-REF-DECLAS, a 6-byte date field which defines the date on which a classified reference is either declassified or reviewed, in format yymmdd.
- p. DATE-REF-DISP, a 6-byte date field which defines the date on which the reference was disposed of, in format yymmdd.
- q. DATE-REF-DSTRD, a 6-byte date field which defines the date on which the reference was destroyed, in format yymmdd.
- r. REF-ARC-HDCY, a 20-byte field which identifies a physical location for the archived hard copy of the reference. This is intended to be a microfiche index location.
- s. REF-ARC-MCFH, a 20-byte field identifying the microfiche sheet on which the reference is located within the above location.
- t. REF-ARC-FRAME, a 3-byte numeric field identifying the frame on the microfiche sheet itself.

#### Organization Record

The organization record supports all types of organizations stored in the database. Its purpose is to provide a common base for all uses of organizations and therefore permit analysis of organizational relationships with other information in the database.

FI-ORGANIZATION, the fixed portion of the organization record contains:

- 1. CLASS-1000, the security classification of the record occurrence, same as SECURITY-CLASS. This field is entered by the IP and

defaults to unclassified if not defined.

2. HANDL-1000, the security releasability code for the record occurrence, same as RELEASABILITY. The field is entered by the IP and defaults to unclassified if not defined.
3. DATE-MOD-1000, the date on which the record occurrence was last modified, in format yymmdd, same as DATE-COMMON. The field is stored automatically by the system.
4. DBID-1000, the database identifier of the record occurrence. The field is stored automatically by the system.
5. LOG-TYPE-1000, the logical type code for the record occurrence, value "AA". The field is stored by the IP.
6. DATE-STORED-1000, the date on which the record was entered into the database, in format yymmdd, same as DATE-COMMON. The field is stored automatically by the system.

ID-ORGANIZATION, the identifier block, contains:

1. ORGAN-ACRONYM, a 16-byte field identifying an organization by an easily recognized acronym.

DA-ORGANIZATION, the data block contains:

1. NAME-ORGAN, a 30-byte organization name field, free form, required.
2. ORGAN-UIC, the unit identification code of the organization, a 6 -character field. A secondary index is defined on this field.
3. NIM-ORGAN-CODE, a 3-character accounting identifier for each organization. A secondary index is defined on this field.
4. ADDR-ORGAN, 6 40-character address lines.

5. STAT-ORGAN, the state in which the organization is located, same as STATE-CODE.
6. ZIP-ORGAN, the postal delivery code for the area in which the organization is located, same as ZIP-CODE.
7. CTRY-ORGAN, the country in which the organization is located, same as COUNTRY-CODE.

#### People Record

The people record serves two primary purposes within the database:

1. A common reference point for persons associated with the database information.
2. A control point for privacy sensitive information.

FI-PEOPLE, the fixed element block of the people record is identical to that of the organization record, with the record identifier of 1003 substituted for 1000.

ID-PEOPLE, the identifier block contains:

1. SOC-SECNO, the social security number, same as SOCIAL-SECURITY. The field is stored by the IP and is mandatory.

DA-PEOPLE, the main data block, contains:

1. NAME-OF-PERSON, the last name, first name, and middle initial(s) of a person, same as NAME-PERSON.
2. ADDR-PER, 6 40-character address lines.
3. STAT-PERSON, the state in which a person resides, same as STATE-CODE.



4. ZIP-PERSON, the postal distribution code defining the postal areas in which a person resides, same as ZIP-CODE.
5. CTRY-PERSON, the country in which a person resides, same as COUNTRY-CODE.

## APPENDIX B.10

## PSL INPUT TAPE FILE FORMAT CONVENTIONS

PSL input provided for loading deliverable machine-readable data to PSL files must be formatted as defined below:

1. EBCDIC code.
2. 9-track, 1600 BPI (800 BPI if 1600 not available).
3. Standard IBM unlabeled format.
4. 80-character fixed-length records.
5. Blocking 1 to 100 as desired.
6. Multiple files on one tape acceptable.

Accompanying the tape as a separate hard copy (or printed on the tape external label) should be the information regarding items 2, 5, and 6 above. If not mentioned, it will be assumed that the tape contains one file with unblocked (block 1) records written at 1600 BPI.

Card images on the file must conform to the following restrictions:

1. No blank card images will appear in the file.
2. Positions 73-80 of each card image will be blank.
3. Positions 1-5 may optionally contain:
  - a. 1-2 contain the version number of the data, initially "01".
  - b. 3 contains a hyphen.
  - c. 4-5 contains the level revision number of the data, initially "01".
4. Cards images containing "/" or "." may not appear in the file.

## APPENDIX B.11

## CRT TERMINAL FUNCTION KEY USE

CRT's for the on-line system will normally be equipped with 12 function keys. These keys are available, as defined below, to provide the user with optional processing support.

Three function keys are reserved:

1. F1 - is the "HELP" key which will be used to call for assistance messages describing the processing function being performed.
2. F2 - terminate function without action and return to transaction handler.
3. F3 - reserved for future use.

Input processing uses the remaining function keys as described below:

1. F4 - perform the input processing function (store, modify, or delete) supported by the IP.
2. F5-12 - request special follow-on processing screens.
3. ENTER/RETURN - for modify/delete, get the data for the requested record and, once data has been retrieved, reject the data and request a new template for another try. For a store, reject the attempt and request a new template for another try.

Output processing (retrieval from the report queue) uses the function keys for:

1. F4 - request the index.
2. F5 - page either the index or the selected report forward.
3. F6 - page either the index or the selected report backward.

4. F7 - request on-line (local) printing of the report.
5. F8 - request off-line (computer printer) printing of the report.
6. F9-12 - special purpose functions.
7. ENTER/RETURN - select the defined report from the report queue.

## APPENDIX C.1

### SINGLE OWNER/MEMBER RECORD ENTITY RELATIONSHIP

The single owner/member entity relationship is the most common and simplest structure within the database. This structure, Figure C1.1, is used where a data entity, occurring once, is accompanied by other entities which may occur multiple times. The owner record occurrence represents the single entity and the member record represents the multiple occurrences of accompanying data entities.

For example: An owner may contain information about a person, such as name and address, which occurs only once. Member records may be defined identifying the jobs the person has held. One occurrence of the member would be stored for each job.

Prior to defining this structure as illustrated, it is necessary that certain questions be asked about the member data:

1. How large is it in total character size?
2. How many occurrences are possible?
3. Can the number of occurrences be limited to a specific quantity without impairing the usability of the data?

The answers to the above questions will determine if the member data can be included within the owner entity as an OCCURS group. This option eliminates pointer overhead and additional processing time during retrieval but places a finite limit on the number of occurrences of the repeating member data which is practical to include.

If the member data is to be shared by another owner entity, it is necessary that a separate member entity be defined.

The UDB physical record structure varies from the logical structure discussed above. From the user's standpoint, however, design of the database is performed in the same manner as if it were conventionally structured.



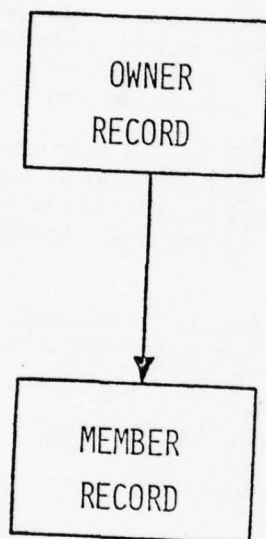


FIGURE C.1.1

## APPENDIX C.2

### BOMP (RECORD OWNS SELF) RELATIONSHIP

The BOMP relationship is an adaptation of the many-to-many relationship in which one record entity can own multiple occurrences of a second record entity while the second can own multiple occurrences of the first. For example: An organization may have many employees. A person employed by the organization may also be employed by other organizations. The primary difference is that the owner entities are the same. The advantage of this structure is that it permits the same record entity to own others of its own kind. (An organization may be composed of many sub-organizations.)

The term "BOMP" is derived from "bill of materials processing" where equipment is described as a series of parts and each part may be made up of one or more other parts. Applied to business processing, it is possible to describe an organization chart by using a relationship such as shown in Figure C.2.1. Organizations are composed of sub-organizations which may continue downward for an unlimited number of hierarchical levels.

The two records illustrated in Figure C.2.1. are the equivalent of the three records shown in Figure C.2.2 with the top and bottom types of this figure identical. The BOMP structure permits an infinite number of subsidiary levels of definition whereas the Figure C.2.2 structure is limited to a single level.

The BOMP structure is the basic physical structure used within the UDB. It may logically simulate all of the other structures permitted by a network database.

FIGURE C.2.1

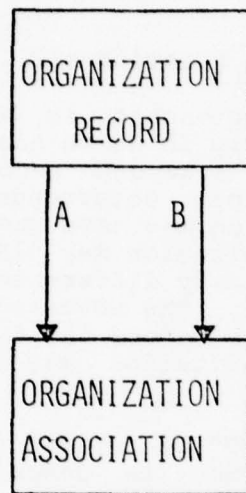
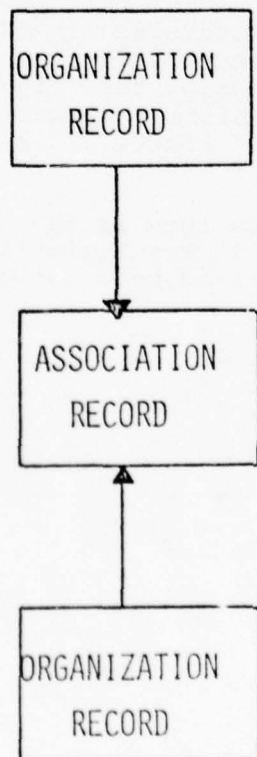


FIGURE C.2.2



APPENDIX D  
DOCUMENTATION STANDARD  
FOR  
ADP SUBSYSTEMS

This appendix is extracted in total from  
DOD Manual 4120.17, December 1972  
Automated Data System Documentation  
Standards Manual

## TABLE OF CONTENTS

SECTION 1	GENERAL
1.1	Purpose of Subsystem Specification
1.2	Project References
SECTION 2	SUMMARY OF REQUIREMENTS
2.1	Subsystem Description
2.2	Subsystem Functions
2.2.1	Accuracy and Validity
2.2.2	Timing
2.3	Flexibility
SECTION 3	ENVIRONMENT
3.1	Equipment Environment
3.2	Support Software Environment
3.3	Interfaces
3.4	Security
3.5	Controls
SECTION 4	DESIGN DATA
4.1	System Logical Flow
4.2	Program Descriptions
4.2.1	Individual Program Description
4.2.1.1	Inputs
4.2.1.2	Outputs
4.2.1.3	Data Base
4.2.2	Program Description



## SECTION 1. GENERAL.

1.1. Purpose of the Subsystem Specification. This paragraph shall describe the purpose of the subsystem specification in the following words, modified when appropriate:

The Subsystem Specification for (subsystem name) is written to fulfill the following objectives:

- a. To provide detailed definition of the subsystem functions.
- b. To communicate details of the on-going analysis to the user's operational personnel.
- c. To define in detail the interfaces with other systems and subsystems and the facilities to be utilized for accomplishing the interface.

1.2. Project References. This paragraph shall provide a brief summary of the references applicable to the history and development of the project. The general nature of the computer programs (tactical, inventory, etc.) to be developed shall be specified. The project sponsor, user, and the operating center(s) that will run the completed computer programs shall be indicated. A list of applicable documents shall be included. At least the following shall be specified, when applicable, by source or author, reference number, title, and security classification:

- a. Function description.
- b. Related system/subsystem specifications.
- c. Any other pertinent documentation or significant correspondence not specified in the Functional Description.

## SECTION 2. SUMMARY OF REQUIREMENTS.

Section 2 of the Subsystem Specification shall provide a summary of the system characteristics and requirements. This section shall be an expansion of the information published in the FD to reflect the determination of additional details. Any changes to the characteristics and requirements set forth in Sections 2 and 3 of the FD must be specifically identified.

2.1. Subsystem Description. This paragraph shall provide a general description of the subsystem to establish a frame of reference for the remainder of the document. Higher order and parallel systems/subsystems and their documentation will be referenced as required to enhance this general description. Included within this description shall be a chart showing the relationship of the user organizations to the major components of the system and a chart showing the interrelationships of the system components for the subsystem. These charts shall be based on or be updated versions of the charts included in paragraph 2.4 of the FD. The more detailed charts to be included in Section 4 shall be based on the charts included in this paragraph.

2.2. Subsystem Functions. This paragraph shall describe the subsystem functions. There will be both qualitative and quantitative descriptions of how the subsystem functions will satisfy the requirements. Although the descriptions of the subsystem functions may be refined and more detailed as a result of the on-going analysis and design, they must maintain a direct relationship to the system functions established in paragraph 2.3 of the FD, and be stated in such a manner that the subsystem environment in Section 3 can be related to them.

2.2.1. Accuracy and Validity. This paragraph shall provide a description of the accuracy requirements imposed on the subsystem. The requirements will be related to paragraph 3.3.1 of the FD. The following accuracy requirements must be considered:

- a. Accuracy requirements of mathematical calculations.
- b. Logical and legal accuracy of alphanumeric data.
- c. Accuracy of transmitted data.

2.2.2. Timing. This paragraph shall provide a description of the timing requirements placed on the subsystem, if they are applicable. The requirements will be related to paragraph 3.1.2 of the FD. The following timing requirements may be considered:

- a. Throughput time.
- b. Response time to queries and to updates of data files.

- c. Response time of major subsystem functions.
- d. Sequential relationship of subsystem functions.
- e. Priorities imposed by types of inputs and changes in modes of operation.
- f. Timing requirements for the range of traffic load under varying operating conditions.
- g. Interleaving requirements for sequencing and interleaving programs and systems (including the requirements for interrupting the operation of a program without loss of data).

2.3. Flexibility. This paragraph shall provide a description of the capability to be incorporated for adapting the subsystem to changing requirements, such as anticipated operational changes, interaction with new or improved systems, and planned periodic changes. Components and procedures designed to be subject to change will be identified.

### SECTION 3. ENVIRONMENT.

This section shall provide an expansion of the environment given in the FD to reflect the additional analysis and changes to the environment. Changes in the environment that do not affect the scope of the project as described in the FD and are the result of on-going analysis and design will be explicitly identified within the appropriate paragraphs of this section. These changes will be discussed in terms of the impacts on the currently available environmental components (equipment, software, etc.) as well as the impacts on the estimates and functions which were based on the original planned environment.

3.1. Equipment Environment. This paragraph shall provide a description of the equipment required for the operation of the subsystem. Included will be descriptions of the equipment presently available as well as a more detailed discussion of the characteristics of any new equipment necessary. Equipment requirements will be related to the requirements

stated in paragraph 4.1 of the FD. A guideline for equipment to be described follows:

- a. Processor(s), including number of each on/off line and size of internal storage.
- b. Storage media, including number of disk units, tape units, etc.
- c. Input/output devices, including number of each on-off line.
- d. Communications net, including line speeds.

3.2. Support Software Environment. This paragraph shall provide a description of the support software with which the computer programs to be developed must interact. Included will be both support software and test software, if needed. The correct nomenclature and documentation references of each such software system, subsystem, and program shall be provided. Included must be a reference to the languages (compiler, assembler, program, query, etc.), the operating system, and any data base management system (DBMS) to be used. This description must relate to and expand on the information provided in paragraph 4.2 of the FD. If operation of the computer programs to be developed is dependent upon forthcoming changes to support software, the nature, status, and anticipated availability date of such changes must be identified and discussed.

3.3. Interfaces. This paragraph shall provide a description of the interfaces with other applications computer programs, including those of other operational capabilities and from other military organizations. The individual interfaces will be related to paragraph 4.3 of the FD. For each interface, the following shall be specified:

- a. Type of interface, such as operator control of a terminal, program interfaces with other programs.
- b. Description of operational implications of data transfer, including security considerations.
- c. Data transfer requirements to and from the subject subsystem and characteristics of communications media/systems used for transfer.

d. Current formats of interchanged data.

e. Interface procedures, including telecommunications considerations.

f. Interface equipment. Interfaces with other subsystems which are to be developed will be described in the same manner.

3.4. Security. This paragraph shall describe the classified components of the subsystem, including computer programs, inputs, outputs, and data bases. These components will be related to paragraph 4.4 of the FD.

3.5. Controls. This paragraph shall provide a presentation of overall subsystem control. Included in this paragraph will be controls such as record counts, batch controls, etc. If no specific controls are to be established at the subsystem level, this will be stated.

#### SECTION 4. DESIGN DATA.

4.1. System Logical Flow. This paragraph shall describe the logical flow of the subsystem. Logical flow of the subsystem will be presented primarily in the form of Macro flowcharts. Flowcharts will be in sufficient detail to permit computer program design. A narrative presentation, when appropriate, will be used to supplement the flowchart. Flowcharts will provide an integrated presentation of the subsystem dynamics, of entrances and exits, and of interfaces with other computer programs. Flowcharts will effectively represent all modes of operations, priorities, cycles, and special handling. The flowcharts will show data flow from input, through the subsystem to the generation of output.

4.2. Program Descriptions. Paragraphs 4.2.1 through 4.2.n shall provide descriptions of the functions (related to paragraph 2.2 of the subsystem specification) of the computer programs in the subsystem.



4.2.1 Program Description. Included in this paragraph will be a description of the inputs, outputs, and data used. Each description will include the information below, as applicable.

4.2.1.1. Inputs. Each input will be described as follows:

- a. Title and tag.
- b. Format and acceptable range of value of each data element.
- c. Number of items (elements).
- d. Means of entry and input initiation procedures; e.g., typewriter, card, tape, sensor, internal.
- e. Expected volume and frequency, including special handling (such as queueing and priority handling) for high density periods.
- f. Priority, e.g., routine, emergency.
- g. Sources, form at source, and disposition of source document.
- h. Security classification of input and individual items.
- i. Requirements for timeliness.

4.2.1.2. Outputs. Each output will be described as follows:

- a. Title and tag.
- b. Format to include headings, line spacing, arrangement, totals, etc.
- c. Number of items.
- d. Preprinted form requirements.
- e. Means of display, e.g., CRT, printer, typewriter, projector, alarm type, internal.

f. Expected volume and frequency including special handling (such as queueing and priority handling) for high density periods.

g. Priority; e.g., routine, emergency.

h. Timing requirements; e.g., response time.

i. Requirements for accuracy.

j. User recipients and use of displays, such as notification, trends, or briefings.

k. Security classification of output and individual items.

4.2.1.3. Data Base. Each data file, table, dictionary, or directory will be described as follows:

a. Title and tag.

b. Description of content.

c. Number of records or entries.

d. Storage, to include type of storage, amount of storage and, if known, beginning and ending addresses.

e. Classification.

f. Data retention.

4.2.2. Program Description. This paragraph (with subsequent paragraphs if required) shall describe the second computer program in the subsystem using the same organization as shown in paragraph 4.2.1.

## APPENDIX E.1

### SECURITY CLASSIFICATIONS

Three types of security information are defined within the data dictionary:

1. ENTRY-SECURITY which defines the classification of data dictionary entries.
2. DATA-SECURITY which defines the classification of data which the dictionary entry defines.
3. OUTPUT-SECURITY which defines the classification of data when presented in the form of an output.

Valid classification attributes are:

1. UNCLASSIFIED.
2. OUO. (Official Use Only)
3. CONFIDENTIAL.
4. SECRET.
5. TOP-SECRET.

The classification attributes are prefixed by ENTRY-, DATA-, or OUTPUT-.

## APPENDIX E.2

### DATA ELEMENT STANDARDS

pply, A number of sources of data element standards are available. These may be divided into descriptive standards and content standards. Descriptive standards are those which are primarily concerned with the definition of the data element and the manner in which it is used. The format used for internal representation of the data may also be defined. Content standards add an additional dimension to descriptive standards. These standards define the legal or valid contents of standard elements, in addition to the structural definition. An example of descriptive standards is dates. The internal representation identifies the element as six characters, in format YYMMDD. This standard definition may become a content standard by adding validity information: month value range 1 through 12; day value range 1 through 31 with conditions based on month.

Every data element defined for the integrated database is subject to existing standards. Table E.2.1. identifies those existing standards which apply to database design. Applicable standards are identified as part of the data element definition process. Where more than one standard applies, all are indicated. If a conflict between standards exists, the conflict must be clearly documented in the element comments section. The conflict will be resolved by the DA staff.

The DA staff has established internal standards of representation for a number of data element types. Table E.2.2. identifies those elements and element types which are subject to these internal standards. The primary reason such standards were adopted is to insure uniform representation of similar data elements internally. This simplifies query processing and program decision logic.

#### APPLICABLE ADP STANDARDS FOR DATA ELEMENTS

1. Federal Information Processing Standards (FIPS).
2. Defense Intelligence Agency Authorized Data Elements and Related Features SO-730-110-76, 1 July 1976 as amended (IDEAS)
3. NAVINTCOM Instruction 5200.2A 12 Feb 1976

Table E.2.1.

#### INTEGRATED DATABASE INTERNAL ELEMENT STANDARDS

##### Element Format Description

Frequency 9(15)v999 This field is primarily associated with radio/ radar frequencies. The measurement is in hertz (Hz) to three decimal places

Time (seconds) 9(7)v999 Second to three decimal places, mS.

Time (hrs,min,sec) 9999 Hours using 24hr clock 99 minutes 0-59 99v999 second to milliseconds.

Table E.2.2.



### APPENDIX E.3

#### LANGUAGE STATEMENT NAMES

Computer programs written in support of the database will be produced in a number of languages depending on the best and most efficient approach. Language use may be limited by convention or regulation. The languages which may be defined are:

1. COBOL. Common Business Oriented Language. This is the most commonly used language for database support and is the standard NIPSSA database development language.
2. FORTRAN. Formula Translator. This language is often used for scientific and statistical programming.
3. PLI. This language support both business and scientific processing.
4. ALC. IBM 360/370 assembly language. This language is used primarily to develop difficult programs which cannot be developed using other available languages.
5. CULPRIT. This is a report and query package which supports the database system. While not truly a language, it is defined as one as far as the data dictionary is concerned.

## APPENDIX E.4

### STANDARD OUTPUT FORMS

Reports may be produced on a variety of printed output forms. Special or preprinted forms should be avoided as turnaround (response to request) time is longer for special forms processing.

The basic form available is used to produce a large percentage of the database reports. This form is 8 1/2 deep by 14 inches wide. It is printed with guide lines spaced every four lines and is printed 8 lines per inch.

Another commonly used form is 11 inches deep by 14 inches wide. It is printed with guide lines spaced every two lines and is printed 6 lines per inch. Several variances of this form are available:

1. Multiple part (carbon) forms in 2, 3, and 4 parts.
2. Reproduction quality single part, unlined.

Additional forms which may be requested are:

1. 8 1/2 inches wide by 11 inches deep, lined or unlined.
2. 14 inches wide by 8 1/2 inches deep, unlined (limited quantity, special order paper).

Users desiring other than standard 8 1/2 high by 14 inch wide, single part, lined forms should plan to supply the volume of paper required to process their reports.

## APPENDIX E.5

### IP FUNCTION OPTIONS

Batch input processing (IP) data contains a data item identifier and a function option code. This code, the fourth position of the data item format, determines what processing will be performed using the data on the input item.

Three basic input processing function codes are used:

1. S - Store a new record occurrence in the database.
2. M - Modify the contents of an existing record occurrence in the database.
2. D - Delete an existing record occurrence from the database.

## APPENDIX E.6

### DECODE TABLE

The DECODE class provides the ability to define special processing of elements within the database. This function is primarily used on output. However, it may be used to identify those data elements which are entered as codes and their conversion arguments. This feature enhances output reporting for the end user.

TABLE-LOOKUP - display the primary table argument for the field and value. These entries are stored in UDB record type 00 (the table record).

DATE-EDIT - Display dates in month, day, and year order rather than year, month, and day as they are stored in the database.

WEEKDAY - Display the day of the week (e.g., Saturday) for a day field instead of the day of the month.

NO-DECODE - This is the default value and is assumed if the DECODE entry is not coded.

## APPENDIX E.7

### PRESENCE TABLE

This table defines the conditions under which a data element must be present in the database or in IP formats.

PRESENCE-REQUIRED - The data element must be present in any IP format where it is used during STORE functions.

PRESENCE-OPTIONAL - The data element may be present in any IP format where it is used.

PRESENCE-QUESTION - The data element is required in any IP format where it is used unless the person entering the data places a question mark (?) in the leftmost position of the data field.

PRESENCE-MANDATORY - The element must be present at all times in the database.



## APPENDIX E.8

### VALIDATION TEST IDENTIFIER TABLES

Every data element present in the integrated database is verified and relocated during input processing. A series of COBOL source library books have been developed to perform validation and transfer functions. A second series of course library books performs validation only. These books are used in the VALID-1 and VALID-2 sections of IP's. The use of these books eliminates large volumes of hand coding of COBOL programs and insures consistent coding of instructions.

The verification and data transfer books are divided into two groups, one for storing data into the database for the first time and one for modifying existing data.

#### STORE VALIDATION AND DATA TRANSFER ENTRIES

	NORMAL	RANGE	DATE	VALUE
	TEST	TEST	TEST	TEST
REQUIRED FIELD NUMERIC	SSZN	SSZP	SSZF	SSZI
ALPHAMERIC	SSZA	SSZC	N/A	SSZI
OPTIONAL FIELD NUMERIC	SSZM	SSZO	SSZE	SSZJ
ALPHAMERIC	SSZG	SSZQ	N/A	SSZJ

#### MODIFY VALIDATION AND DATA TRANSFER ENTRIES

	NORMAL	RANGE	DATE	VALUE
	TEST	TEST	TEST	TEST
REQUIRED FIELD NUMERIC	SMZV	SMZU	SMZC	SMZI
ALPHAMERIC	SMZB	SMZF	N/A	SMZI
OPTIONAL FIELD NUMERIC	SMZW	SMZY	SMZX	SMZH
* DELETE ALLOWED				
ALPHAMERIC	SMZA	SMZE	N/A	SMZH
OPTIONAL FIELD NUMERIC	SMZM	SMZU	SMZC	SMZI
NO * DELETE ALLOWED				
ALPHAMERIC	SMZB	SMZF	N/A	SMZI

## VALIDATION ONLY ENTRIES

Source book names for the validation only functions are coded to identify the specific validation to be performed. The book name is four characters long. Character definitions are:

1. First character. The first character defines the database function being performed:

- a. S - Store
- b. M - Modify
- c. D - Delete.

2. Second character. The second character defines whether the book will test for numeric or alphanumeric fields:

- a. A - Alphanumeric
- b. N - Numeric.

3. Third character. The third character defines whether a value must be present in the field being tested:

- a. O - Optional
- b. R - Required
- c. Q - Required, but null allowed if Q in the first position
- d. A - Asterisk in first position permits field contents to be deleted in database record
- e. K - Field is used as a database currency key and is required.

4. Fourth character. The fourth character defines special validation functions to be performed on the field being tested:

- a. N - Normal, no special tests
- b. R - Range test, inclusive values

- c. T - Table lookup for specific value
- d. D - Date validation, 6-digit date (yyymmdd)
- e. E - Date validation, 4-digit date (yy-mm).

## APPENDIX E.9

### DATA-SUBJECT TABLE

Each data element will be identified with one or more data subjects. This appendix provides a list of current subjects. Additional subjects may be added with the approval of the DBA.

ACOUSTIC INTELLIGENCE  
ADMINISTRATIVE  
AIR WARFARE  
AIRCRAFT INFORMATION  
AMMO AND EXPLOSIVES  
ANTI-SUB WARFARE

BIOGRAPHIC  
BIOLOGICAL

COLLECTION AND DISSEMINATION  
COMMON-ELEMENTS  
COMMON-RECORDS  
COMMUNICATION  
COMMUNICATION INTELLIGENCE  
COMPUTER OPERATIONS  
COMPUTER SOFTWARE  
CORRESPONDENCE

DEFINITION  
DOCUMENT MANAGEMENT

ELECTRO-OPTIC INTELLIGENCE  
ELECTRONIC INTELLIGENCE  
ELECTRONIC SYSTEMS DATA  
ELECTRONIC WARFARE  
ENVIRONMENTAL

FINANCIAL MANAGEMENT

GEOGRAPHY AND NAVIGATION INFO  
GEOPOLITICAL

IMAGERY  
INSTALLATION  
INTELLIGENCE

LIBRARY  
LOGISTICS

MAIL  
MAPPING AND CHARTING  
MEDICAL DATA  
MILITARY AND CIVILIAN ORG. INFO.  
MISCELLANEOUS  
MISSILE INFORMATION

NON-NUCLEAR WEAPONS DATA  
NUCLEAR INTELLIGENCE  
NUCLEAR WEAPONS DATA

ORDERS OF BATTLE  
ORGANIZATIONS

PERSONNEL  
PETROL/OIL/LUBRICANTS INFORMATION  
PLATFORMS  
PROJECT MANAGEMENT

RADAR INTELLIGENCE  
RADIATION INTELLIGENCE  
REFERENCE

SECURITY CLASSIFICATION  
SECURITY, GENERAL  
SHIP INFORMATION  
SHORE FACILITIES  
SPACE TECHNOLOGY DATA  
SPECIAL SYSTEM  
STRATEGIC PLANNING  
SUPPLY DATA  
SYSTEM  
SYSTEMS ANALYSIS/PROGRAMMING

TACTICAL INFORMATION  
TARGET INFORMATION  
TELEMETRY  
TIME DATA  
TRAINING  
TRANSPORTATION  
UDB

WEATHER DATA  
WRECK DATA



## APPENDIX E.10

### ELEMENT DESIGNATORS

The data dictionary class ELEMENT DESIGNATOR is used to identify certain conditions related to data elements and element groups. The designators are specifically used to support automatic input processing (IP) module generation and on-line output display definition. Each designator MUST be prefixed by "DES-".

Valid designators are:

1. FIXED. The data element is fixed length and will always contain the maximum number of characters. For example, security clearance is one character long and will always contain the one character. A country code is two characters long and always contains two characters of data.
2. VARIABLE. The data element is of fixed or variable length but may contain a variable number of data characters. For example, a person's name is 27 characters long, but the name of a single person may only occupy 21 of the available positions.
3. KEY. The data element is used as the access key or one of the fields in an access key for a record occurrence stored in the database.
4. INDEX. The data element is used as a secondary index field or as one of a group of fields addressed by the secondary index.
5. SECURE. The element contents will not be displayed on output reports except where specifically authorized. This feature is useful for passwords or very sensitive data elements which must be carefully controlled.
6. TEXT. The data element contains free-form textual material in both upper and lower case.
7. SUBSCRIPT. The data element is used as a

subscript.

8. NO-VERIFY. The data element does not require verification. This designator is used for alphameric optional elements where verification is impractical or unnecessary.
9. ENCRYPT. The data element is to be encrypted to further protect its contents. This designator should be avoided except where data is extremely sensitive as significant overhead is created.
10. CODE. The data element is a code which may be expanded into another or larger definition, e.g., a state code where VA means Virginia. Where this designator is used, UDB table records must be present which defines the code values and their expansions for the element.
11. EXPAND. This designator is used only for data elements defined as part of a display description. Where defined, the code found within the database element occurrence is expanded into its larger definition for display purposes.
12. HIGHLIGHT. This designator is used only for data elements defined as part of an on-line display description. Where defined, the displayed element value is shown as high-intensity, if available, or blinking.
13. DARK. This designator is used only for data elements defined as part of an on-line display description. Where defined, the designated element value is not displayed. Do not confuse this designator with SECURE. SECURE affects any use of the element for all types of outputs. DARK applies only to the specific output display where it is used.
14. DATE. This designator is used to identify group elements which consist of a 6-digit date in the format yymmdd.

AD-A069 868

NAVAL INTELLIGENCE PROCESSING SYSTEMS SUPPORT ACTIVIT--ETC F/6 9/2  
INTEGRATED DATABASE DEVELOPMENT AND DESIGN GUIDE. VERSION 2.0.(U)  
MAY 79

UNCLASSIFIED

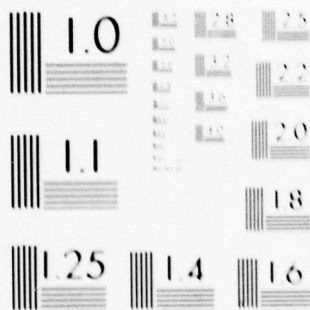
NIPSSA-UM-8000/2.0-5/79

NL

3 OF 7

AD  
A069868





MICROCOPY RESOLUTION TEST CHART  
NATIONAL BUREAU OF STANDARDS-1963-A

15. DATE4. This designator is used to identify group elements which consist of a 4-digit date in the format yymm.



## APPENDIX E.11

### PREDEFINED DATA ELEMENTS

Some basic data elements repeat throughout the database continuously. These are elements which have identical structure and similar content but have completely different meaning and purpose within the database.

This appendix lists those common elements which have been defined within the data dictionary. These elements may be used as objects of the SAME AS ELEMENT clause to reduce coding and improve the consistency of definition among commonly structured elements.

1. MONTH. A two-digit month of the year, with a valid value range of 1 through 12.
2. DAY. A two-digit day of the month, with a valid value range of 1 through 31, dependent upon the associated month.
3. YEAR. A two-digit year of the century, with a valid value range of 0 through 99.
4. DATE-COMMON. A group element composed of YEAR, MONTH, and DAY.
5. TIME-HOUR. A 2-digit hour of the day, with a valid value range of 0 through 23.
6. TIME-MINUTE. A 2-digit minute within an hour, with a valid value range of 0 through 59.
7. TIME-SECOND. A 2-digit second within an hour, with a valid value range of 0 through 59.
8. TIME-ZONE. A single-character zone identifier, defining the time zone where the time reading took place.
9. TIME-COMPOSIT. A group element composed of TIME-HOUR, TIME-MINUTE, TIME-SECOND, and TIME-ZONE.

10. SECURITY-CLASS. A one-character field containing standard security classification information.
11. RELEASABILITY. A two-character field containing standard security releasability codes as defined in DIAM65-19.
12. COUNTRY-CODE. A two-character field contains standard country identifiers defined by DIA.
13. ZIP-CODE. A five-digit postal zip code identifier.
14. STATE-CODE. A two-character field containing standard FIPS state identifier codes.
15. NAME-ORGAN. A 30-character field identifying the name of an organization.
16. ORGAN-ACRONYM. A 16-character field identifying an acronym used as alternate identifier of an organization.
17. SOCIAL-SECURITY. A 11-digit field defining the social security number of a person.
18. FREQ-VALUE. A 18-digit field containing any value which represents the use of frequency measured in Hertz. The field has 15 significant positions and 3 decimal places, measuring to milli-Hertz.
19. DATE-JULIAN. A 3-digit field representing the Julian date value, with a valid range value of 1 through 366. NOTE: Where the Julian date is resident within the database, standards require the the Gregorian representation of the same date be present also.
20. TIME-OF-DAY. A group element composed of TIME-HOUR and TIME-MINUTE.
21. NAME-PERSON. A 27-character data element

which contains the name of a person. The last name is entered first, left justified. It is followed by a comma and the first name/initial(s) as desired.

## APPENDIX E.11A

### PREDEFINED DATA ELEMENTS (IDD SYNTAX)

This subset to Appendix E.11 provides the detailed IDD syntax for each of the predefined data elements shown in the appendix. The elements are stored in the data dictionary exactly as described below. Any descriptive clause may be altered during use of the element in the SAME AS clause by entering the replacement clause as part of the definition of the mirrored element.

ADD ELEMENT NAME IS MONTH  
PREPARED BY LET  
ELEMENT DESCRIPTION IS  
'MONTH OF THE YEAR'  
ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
DATA-SECURITY IS DATA-UNCLASSIFIED  
APPLICATION-SYSTEM IS UDB  
PICTURE IS 99  
USAGE IS DISPLAY  
VALUE IS ZEROS  
STORE-VALIDATION IS SSZM  
MODIFY-VALIDATION IS SMZD  
ELEMENT DESIGNATOR IS DES-FIXED  
PRESENCE IS PRESENCE-OPTIONAL  
ELEMENT DEFINITION IS  
'A TWO-DIGIT MONTH OF THE YEAR'  
DATA-SUBJECT IS COMMON-ELEMENTS  
USER IS 'NIPSSA03 TOWNER'  
RESPONSIBLE FOR DEFINITION  
RANGE IS '01' THRU '12'.



ADD ELEMENT NAME IS DAY  
PREPARED BY LET  
ELEMENT DESCRIPTION IS  
'DAY OF THE MONTH'  
ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
DATA-SECURITY IS DATA-UNCLASSIFIED  
APPLICATION-SYSTEM IS UDB  
PICTURE IS 99  
USAGE IS DISPLAY  
VALUE IS ZEROS  
STORE-VALIDATION IS SSZM  
MODIFY-VALIDATION IS SMZD  
ELEMENT DESIGNATOR IS DES-FIXED  
PRESENCE IS PRESENCE-OPTIONAL  
ELEMENT DEFINITION IS  
'A TWO-DIGIT DAY OF THE MONTH'  
DATA-SUBJECT IS COMMON-ELEMENTS  
USER IS 'NIPSSA03 TOWNER'  
RESPONSIBLE FOR DEFINITION  
RANGE IS '01' THRU '31'  
COMMENTS  
'RANGE VARIES BY MONTH OF YEAR'.

ADD ELEMENT NAME IS YEAR  
PREPARED BY LET  
ELEMENT DESCRIPTION IS  
'YEAR OF THE CENTURY'  
ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
DATA-SECURITY IS DATA-UNCLASSIFIED  
APPLICATION-SYSTEM IS UDB  
PICTURE IS 99  
USAGE IS DISPLAY  
VALUE IS ZEROS  
STORE-VALIDATION IS SSZM  
MODIFY-VALIDATION IS SMZD  
ELEMENT DESIGNATOR IS DES-FIXED  
PRESENCE IS PRESENCE-OPTIONAL  
ELEMENT DEFINITION IS  
'THE YEAR WITHIN A CENTURY'  
DATA-SUBJECT IS COMMON-ELEMENTS  
USER IS 'NIPSSA03 TOWNER'  
RESPONSIBLE FOR DEFINITION  
RANGE IS '00' THRU '99'.



ADD ELEMENT NAME IS DATE-COMMON  
PREPARED BY LET  
ELEMENT DESCRIPTION IS  
'STANDARD DATE GROUP'  
ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
DATA-SECURITY IS DATA-UNCLASSIFIED  
APPLICATION-SYSTEM IS UDB  
ELEMENT DESIGNATOR IS DES-DATE  
PRESENCE IS PRESENCE-OPTIONAL  
ELEMENT DEFINITION IS  
'DATE COMPOSED OF YYMMDD'  
DATA-SUBJECT IS COMMON-ELEMENTS  
USER IS 'NIPSSA03 TOWNER'  
RESPONSIBLE FOR DEFINITION  
COMMENTS  
'FORMAT YYMMDD'  
SUBORDINATE ELEMENTS ARE  
YEAR  
MONTH  
DAY.

ADD ELEMENT NAME IS TIME-HOUR  
PREPARED BY LET  
ELEMENT DESCRIPTION IS  
'HOUR OF THE DAY'  
ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
DATA-SECURITY IS DATA-UNCLASSIFIED  
APPLICATION-SYSTEM IS UDB  
PICTURE IS 99  
USAGE IS DISPLAY  
VALUE IS ZEROS  
STORE-VALIDATION IS SSZM  
MODIFY-VALIDATION IS SMZD  
ELEMENT DESIGNATOR IS DES-FIXED  
PRESENCE IS PRESENCE-OPTIONAL  
ELEMENT DEFINITION IS  
'THE HOUR WITHIN A DAY USING A 24-HOUR CLOCK'  
DATA-SUBJECT IS COMMON-ELEMENTS  
USER IS 'NIPSSA03 TOWNER'  
RESPONSIBLE FOR DEFINITION  
RANGE IS '00' THRU '23'.

ADD ELEMENT NAME IS TIME-MINUTE  
PREPARED BY LET  
ELEMENT DESCRIPTION IS  
'MINUTE OF AN HOUR'  
ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
DATA-SECURITY IS DATA-UNCLASSIFIED  
APPLICATION-SYSTEM IS UDB  
PICTURE IS 99  
USAGE IS DISPLAY  
VALUE IS ZEROS  
STORE-VALIDATION IS SSZM  
MODIFY-VALIDATION IS SMZD  
ELEMENT DESIGNATOR IS DES-FIXED  
PRESENCE IS PRESENCE-OPTIONAL  
ELEMENT DEFINITION IS  
'MINUTES WITHIN AN HOUR'  
DATA-SUBJECT IS COMMON-ELEMENTS  
USER IS 'NIPSSA03 TOWNER'  
RESPONSIBLE FOR DEFINITION  
RANGE IS '00' THRU '59'.

ADD ELEMENT NAME IS TIME-SECOND  
PREPARED BY LET  
ELEMENT DESCRIPTION IS  
'SECONDS WITHIN A MINUTE'  
ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
DATA-SECURITY IS DATA-UNCLASSIFIED  
APPLICATION-SYSTEM IS UDB  
PICTURE IS 99  
USAGE IS DISPLAY  
VALUE IS ZEROS  
STORE-VALIDATION IS SSZM  
MODIFY-VALIDATION IS SMZD  
ELEMENT DESIGNATOR IS DES-FIXED  
PRESENCE IS PRESENCE-OPTIONAL  
ELEMENT DEFINITION IS  
'SECONDS WITHIN A MINUTE'  
DATA-SUBJECT IS COMMON-ELEMENTS  
USER IS 'NIPSSA03 TOWNER'  
RESPONSIBLE FOR DEFINITION  
RANGE IS '00' THRU '59'.

ADD ELEMENT NAME IS TIME-ZONE  
PREPARED BY LET  
ELEMENT DESCRIPTION IS  
'GEOGRAPHICAL TIME ZONE'  
ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
DATA-SECURITY IS DATA-UNCLASSIFIED  
APPLICATION-SYSTEM IS UDB  
PICTURE IS X  
USAGE IS DISPLAY  
VALUE IS SPACES  
STORE-VALIDATION IS SSZG  
MODIFY-VALIDATION IS SMZA  
ELEMENT DESIGNATOR IS DES-FIXED  
PRESENCE IS PRESENCE-OPTIONAL  
ELEMENT DEFINITION IS  
'THE ZONE IN WHICH THE TIME IS MEASURED'  
DATA-SUBJECT IS COMMON-ELEMENTS  
USER IS 'NIPSSA03           TOWNER'  
RESPONSIBLE FOR DEFINITION.

ADD ELEMENT NAME IS TIME-COMPOSIT  
PREPARED BY LET  
ELEMENT DESCRIPTION IS  
'TIME OF THE DAY'  
ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
DATA-SECURITY IS DATA-UNCLASSIFIED  
APPLICATION-SYSTEM IS UDB  
ELEMENT DESIGNATOR IS DES-NO-VERIFY  
PRESENCE IS PRESENCE-OPTIONAL  
ELEMENT DEFINITION IS  
'TIME OF THE DAY WITH THE RELEVANT TIME ZONE'  
DATA-SUBJECT IS COMMON-ELEMENTS  
USER IS 'NIPSSA03           TOWNER'  
RESPONSIBLE FOR DEFINITION  
SUBORDINATE ELEMENTS ARE  
TIME-HOUR  
TIME-MINUTE  
TIME-SECOND  
TIME-ZONE.

ADD ELEMENT NAME IS TIME-OF-DAY  
PREPARED BY LET  
ELEMENT DESCRIPTION IS  
'TIME OF THE DAY'  
ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
DATA-SECURITY IS DATA-UNCLASSIFIED  
APPLICATION-SYSTEM IS UDB  
ELEMENT DESIGNATOR IS DES-NO-VERIFY  
PRESENCE IS PRESENCE-OPTIONAL  
ELEMENT DEFINITION IS  
'TIME OF THE DAY WITHOUT TIME ZONE'  
DATA-SUBJECT IS COMMON-ELEMENTS  
USER IS 'NIPSSA03            TOWNER'  
RESPONSIBLE FOR DEFINITION  
SUBORDINATE ELEMENTS ARE  
TIME-HOUR  
TIME-MINUTE  
TIME-SECOND.

ADD ELEMENT NAME IS SECURITY-CLASS  
PREPARED BY LET  
ELEMENT DESCRIPTION IS  
'SECURITY CLASSIFICATION'  
ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
DATA-SECURITY IS DATA-UNCLASSIFIED  
APPLICATION-SYSTEM IS UDB  
PICTURE IS X  
USAGE IS DISPLAY  
VALUE IS SPACES  
STORE-VALIDATION IS SSZG  
MODIFY-VALIDATION IS SMZA  
ELEMENT DESIGNATOR IS DES-CODE  
PRESENCE IS PRESENCE-OPTIONAL  
ELEMENT DEFINITION IS  
'STANDARD SECURITY CLASSIFICATION CODE'  
DATA-SUBJECT IS COMMON-ELEMENTS  
USER IS 'NIPSSA03            TOWNER'  
RESPONSIBLE FOR DEFINITION  
RANGE IS 'U'  
RANGE IS 'O'  
RANGE IS 'C'  
RANGE IS 'S'  
RANGE IS 'T'  
RANGE IS ' '  
COMMENTS  
'BLANK IS ASSUMED UNCLASSIFIED'.



ADD ELEMENT NAME IS RELEASABILITY  
 PREPARED BY LET  
 ELEMENT DESCRIPTION IS  
 'SECURITY RELEASABILITY'  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-UNCLASSIFIED  
 APPLICATION-SYSTEM IS UDB  
 PICTURE IS XX  
 USAGE IS DISPLAY  
 VALUE IS SPACES  
 STORE-VALIDATION IS SSZG  
 MODIFY-VALIDATION IS SMZA  
 ELEMENT DESIGNATOR IS DES-FIXED  
 PRESENCE IS PRESENCE-OPTIONAL  
 ELEMENT DEFINITION IS  
 'THE RELEASABILITY CODE FOR A SECURITY CLASS'  
 - 'CONFORMS TO DIAM 65-19'  
 DATA-SUBJECT IS COMMON-ELEMENTS  
 USER IS 'NIPSSA03 TOWNER'  
 RESPONSIBLE FOR DEFINITION.

ADD ELEMENT NAME IS COUNTRY-CODE  
 PREPARED BY LET  
 ELEMENT DESCRIPTION IS  
 'STANDARD COUNTRY IDENTIFIER'  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-UNCLASSIFIED  
 APPLICATION-SYSTEM IS UDB  
 PICTURE IS XX  
 USAGE IS DISPLAY  
 VALUE IS SPACES  
 STORE-VALIDATION IS SSZG  
 MODIFY-VALIDATION IS SMZA  
 ELEMENT DESIGNATOR IS DES-CODE  
 PRESENCE IS PRESENCE-OPTIONAL  
 ELEMENT DEFINITION IS  
 'THE CODE WHICH IDENTIFIES COUNTRIES OF THE WORLD'  
 DATA-SUBJECT IS COMMON-ELEMENTS  
 USER IS 'NIPSSA03 TOWNER'  
 RESPONSIBLE FOR DEFINITION  
 COMMENTS  
 'MEETS DIA STANDARDS'.



ADD ELEMENT NAME IS ZIP-CODE  
PREPARED BY LET  
ELEMENT DESCRIPTION IS  
'US POSTAL ZIP CODE'  
ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
DATA-SECURITY IS DATA-UNCLASSIFIED  
APPLICATION-SYSTEM IS UDB  
PICTURE IS 99999  
USAGE IS DISPLAY  
VALUE IS ZEROS  
STORE-VALIDATION IS SSZM  
MODIFY-VALIDATION IS SMZD  
ELEMENT DESIGNATOR IS DES-FIXED  
PRESENCE IS PRESENCE-OPTIONAL  
ELEMENT DEFINITION IS  
'US POSTAL DISTRIBUTION CODE'  
DATA-SUBJECT IS COMMON-ELEMENTS  
USER IS 'NIPSSA03           TOWNER'  
RESPONSIBLE FOR DEFINITION.

ADD ELEMENT NAME IS STATE-CODE  
PREPARED BY LET  
ELEMENT DESCRIPTION IS  
'STANDARD STATE IDENTIFIER'  
ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
DATA-SECURITY IS DATA-UNCLASSIFIED  
APPLICATION-SYSTEM IS UDB  
PICTURE IS XX  
USAGE IS DISPLAY  
VALUE IS SPACES  
STORE-VALIDATION IS SSZG  
MODIFY-VALIDATION IS SMZA  
ELEMENT DESIGNATOR IS DES-CODE  
PRESENCE IS PRESENCE-OPTIONAL  
ELEMENT DEFINITION IS  
'FIPS STANDARD CODE FOR US STATES'  
DATA-SUBJECT IS COMMON-ELEMENTS  
USER IS 'NIPSSA03           TOWNER'  
RESPONSIBLE FOR DEFINITION.

ADD ELEMENT NAME IS NAME-ORGAN  
PREPARED BY LET  
ELEMENT DESCRIPTION IS  
'NAME OF AN ORGANIZATION'  
ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
DATA-SECURITY IS DATA-UNCLASSIFIED  
APPLICATION-SYSTEM IS UDB  
PICTURE IS X(30)  
USAGE IS DISPLAY  
VALUE IS SPACES  
STORE-VALIDATION IS SSZG  
MODIFY-VALIDATION IS SMZA  
ELEMENT DESIGNATOR IS DES-VARIABLE  
PRESENCE IS PRESENCE-OPTIONAL  
ELEMENT DEFINITION IS  
'NAME OF AN ORGANIZATION'  
DATA-SUBJECT IS COMMON-ELEMENTS  
USER IS 'NIPSSA03           TOWNER'  
RESPONSIBLE FOR DEFINITION  
COMMENTS  
'FREE-FORM'.

ADD ELEMENT NAME IS ORGAN-ACRONYM  
PREPARED BY LET  
ELEMENT DESCRIPTION IS  
'ORGANIZATION ABBREVIATION'  
ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
DATA-SECURITY IS DATA-UNCLASSIFIED  
APPLICATION-SYSTEM IS UDB  
PICTURE IS X(16)  
USAGE IS DISPLAY  
VALUE IS SPACES  
STORE-VALIDATION IS SSZG  
MODIFY-VALIDATION IS SMZA  
ELEMENT DESIGNATOR IS DES-VARIABLE  
PRESENCE IS PRESENCE-OPTIONAL  
ELEMENT DEFINITION IS  
'SHORT NAME OR ACRONYM IDENTIFYING AN ORGANIZATION'  
DATA-SUBJECT IS COMMON-ELEMENTS  
USER IS 'NIPSSA03           TOWNER'  
RESPONSIBLE FOR DEFINITION.

ADD ELEMENT NAME IS SOCIAL-SECURITY  
 PREPARED BY LET  
 ELEMENT DESCRIPTION IS  
 'US SOCIAL SECURITY NUMBER'  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-UNCLASSIFIED  
 APPLICATION-SYSTEM IS UDB  
 PICTURE IS X(11)  
 USAGE IS DISPLAY  
 VALUE IS SPACES  
 STORE-VALIDATION IS SSZG  
 MODIFY-VALIDATION IS SMZA  
 ELEMENT DESIGNATOR IS DES-NO-VERIFY  
 PRESENCE IS PRESENCE-OPTIONAL  
 ELEMENT DEFINITION IS  
 'THE SOCIAL SECURITY NUMBER, RIGHT  
 - 'JUSTIFIED'  
 DATA-SUBJECT IS COMMON-ELEMENTS  
 USER IS 'NIPSSA03 TOWNER'  
 RESPONSIBLE FOR DEFINITION.

ADD ELEMENT NAME IS FREQ-VALUE  
 PREPARED BY LET  
 ELEMENT DESCRIPTION IS  
 'FREQUENCY VALUE'  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-UNCLASSIFIED  
 APPLICATION-SYSTEM IS UDB  
 PICTURE IS 9(15)V999  
 USAGE IS DISPLAY  
 VALUE IS ZEROS  
 STORE-VALIDATION IS SSZM  
 MODIFY-VALIDATION IS SMZD  
 ELEMENT DESIGNATOR IS DES-FIXED  
 PRESENCE IS PRESENCE-OPTIONAL  
 ELEMENT DEFINITION IS  
 'THE FIELD REPRESENTS FREQUENCY (NORMALLY HERTZ)'  
 - 'TO 15 SIGNIFICANT DIGITS AND THOUSANDTHS'  
 DATA-SUBJECT IS COMMON-ELEMENTS  
 USER IS 'NIPSSA03 TOWNER'  
 RESPONSIBLE FOR DEFINITION  
 COMMENTS  
 'ASSUMED DECIMAL POINT 3 POSITIONS FROM'  
 - 'RIGHT OF FIELD. MEASURES TO MILLIHERTZ'.

ADD ELEMENT NAME IS DATE-JULIAN  
PREPARED BY LET  
ELEMENT DESCRIPTION IS  
'JULIAN DATE IN YEAR'  
ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
DATA-SECURITY IS DATA-UNCLASSIFIED  
APPLICATION-SYSTEM IS UDB  
PICTURE IS 999  
USAGE IS DISPLAY  
VALUE IS ZEROS  
STORE-VALIDATION IS SSZM  
MODIFY-VALIDATION IS SMZD  
ELEMENT DESIGNATOR IS DES-FIXED  
PRESENCE IS PRESENCE-OPTIONAL  
ELEMENT DEFINITION IS  
'THE JULIAN DATE OF A DAY IN A YEAR'  
DATA-SUBJECT IS COMMON-ELEMENTS  
USER IS 'NIPSSA03 TOWNER'  
RESPONSIBLE FOR DEFINITION  
RANGE IS '001' THRU '366'.

## APPENDIX E.12

### FREQUENCY OF SERVICE CODES

Service analyses contain a criteria defining the frequency with which the user desires the service rendered. This appendix provides a list of standard service definition codes. Multiple frequency codes may be used if the service is required at more than one frequency, such as adhoc and monthly.

Where the defined codes do not fit the user's service requirements, special codes may be created and used. Special codes must be enclosed in single quote marks.

1. ADHOC. The service may be requested at any time.
2. DAILY. The service is performed on a scheduled basis every day.
3. WEEKLY. The service is performed on a scheduled basis once every week.
4. BIWEEKLY. The service is performed on a scheduled basis once every two weeks.
5. MONTHLY. The service is performed on a scheduled basis once every month.
6. BIMONTHLY. The service is performed on a scheduled basis once every two months.
7. QUARTERLY. The service is performed on a scheduled basis once every quarter.
8. SEMIANNUAL. The service is performed on a scheduled basis once every six months.
9. ANNUALLY. The service is performed on a scheduled basis once every calendar year.



## APPENDIX E.13

### PRIORITY OF SERVICE CODES

As part of a service analysis, each service defined is assigned a priority for development. This appendix defines a standard list of priority codes to be used.

1. AS-AVAILABLE. The service is to be developed when all other service requests for the application have been satisfied and as resources are available.
2. LOW-PRIORITY. The service is to be developed when all other service requests of higher priority have been satisfied.
3. NORMAL-PRIORITY. The service is to be developed as normally scheduled after requests of higher priority have been satisfied. A majority of service requests should fall in this category.
4. HIGH-PRIORITY. The service analysis should be developed as one of the first efforts in the application.
5. URGENT-PRIORITY. The service must be developed as soon as possible.
6. FUTURE-NEED. The service request defines a requirement which will not be needed for at least six months.

## APPENDIX E.14

### RESPONSE TIME OF SERVICE CODES

The service analysis includes the definition of the time frame the user desires to receive a response to a request for a specific production service. The codes in this appendix define standard response time values. Where an unusual response time requirement exists, it may be defined and entered enclosed in single quote marks.

1. IMMEDIATE-RESPONSE. The user desires response to the service request in real-time or near real-time (not to exceed two minutes).
2. RESPONSE-1. The user desires response to the service request within one hour.
3. RESPONSE-6. The user desires response to the service request within 6 hours.
4. RESPONSE-12. The user desires response to the service request within 12 hours.
5. OVERNIGHT-RESPONSE. The user desires response to the service request by the beginning of the following work day.

## APPENDIX E.15

### SERVICE HISTORY CODES

The service history attribute adds an understanding to the service request of the history of such services to the user. Where the user has been receiving ADP services, the history may show that this service request is an extension or upgrade of an existing request and provide the designer with a source of background information.

1. EXISTING-SERVICE. The requested service is existing in either ADP or manual form.
2. NEW-SERVICE. The requested service has not been produced previously.
3. PLANNED-SERVICE. The service is part of an overall development plan but will not be implemented in the near future (within 6 months).
4. UPGRADED-SERVICE. The request is for an upgrade or enhancement to an existing user service.

## APPENDIX E.16

### SERVICE OUTPUT MODES

Service requests which result in the preparation of outputs from the database are generally classified by the mode codes in this appendix.

1. REPORT-MODE. The service request results in a printed report.
2. DISPLAY-MODE. The service request results in a CRT display.
3. CARD-MODE. The service request results in punched card output.
4. TAPE-MODE. The service request results in magnetic tape output.
5. FICHE-MODE. The service request results in microfiche output.

## APPENDIX E.17

### STANDARD TP-IP FUNCTION COMMAND CODES

It is desirable to provide the capability to perform additional functional transaction programs as a result of either a user operator decision or program logic determination. The codes shown below are used in the "\$M" parameter card of the on-line IP program generator to provide this flexibility.

Valid codes are:

1. F1 through F9, corresponding to functions keys 1 through 9.
2. FA through FC, corresponding to function keys 10 through 12.
3. A1 through A3, corresponding to PA keys 1 through 3.
4. EN, corresponding to the ENTER/RETURN key.
5. XI, corresponding to the detection of an "information" level message created by the IP. This exit should be used with extreme care because information level messages are very common.
6. XW, corresponding to the detection of a "warning" level message created by the IP. As with the "information" level message, the number of warning messages generated by an IP dictates that use of this option be carefully reviewed.
7. XC, corresponding to the "correctable" level error message created by the IP.
8. XE, corresponding to the "fatal" level error message created by the IP.
9. XD, corresponding to the "disaster" level error message created by the IP.



10. X1 through X9, corresponding to special exits which the IP may invoke.

## APPENDIX F

### SAMPLE USERS GUIDES

This appendix contains portions of two users guides. The first is the Users Guide for the Generalized Working-file Database Facility. This guide illustrates the general format of the overall approach to be followed when developing any users guide.

The second illustration is a portion of the NIMIS users guide which provides more examples of the instructions for batch input processing data preparation.

It is the intent of the users guide to provide an easily understood tool for end users of a database application. Simplicity and clarity are the primary guidelines. Examples should be shown where applicable.

### APPENDIX G

### INTEGRATED DATA DICTIONARY (IDD) SYNTAX

This appendix contains excerpts from the users manual for the Integrated Data Dictionary (IDD) developed and marketed by Cullinane Corporation of Boston.

The material in this appendix is copyrighted by Cullinane Corporation and used with their permission.

8331-UM-02.0/2-79

USERS GUIDE  
FOR THE  
GENERAL WORKING-FILE DATABASE FACILITY  
SUPPORTING  
THE NAVAL INTELLIGENCE COMMAND  
THROUGH THE  
NAVAL INTELLIGENCE COMMAND ON-LINE SYSTEM (NICOLS)

VERSION 2.0  
FEBRUARY 1979

PREPARED BY  
NAVAL INTELLIGENCE PROCESSING SYSTEM SUPPORT ACTIVITY  
(NIPSSA)

# RECORD OF CHANGES

- 1.1 7/5/78 Cover Page, Table of Contents, 3.3, 3.12, 3.17, 3.21, 4.1, 4.4, 4.6, Record Definition Form, Appendix B.2
- 2.0 2/8/79 Cover Page, Table of Contents, 6.1 thru 6.37

## FOREWORD

This Users Guide is designed to be used as a quick reference to assist you in defining automated working files. Each function is prepared separately with an illustration. The remainder of the Guide has been organized with the pages horizontal instead of vertical to make the instruction format more effective.



(2.0)

TABLE OF CONTENTS

1.	INTRODUCTION	
1.1.	Purpose	3.2.3. File Deletion
2.	CAPABILITY DESCRIPTION	4. QUERY AND REPORT PREPARATION
2.1.	Facility Capabilities	5. TELEPROCESSING INPUT (TBA)
3.	CREATING AND MAINTAINING THE WORKING-FILE	6. TELEPROCESSING OUTPUT
3.1.	Overview	6.1. Overview
3.2.	File Creation and Maintenance Functions	6.2. PART I - Generate CULPRIT Reports From Working-File
3.2.1.	File Creation	a. Change 1
		b. Change 2
	a. Identification of the file user to NICOLS.	6.3. CULPRIT Retrieval Errors
	b. Identification of the file control information to NICOLS.	6.4. PART II - Display CULPRIT Output On Terminals
	c. Definition of file records for maintenance.	6.5. Errors from Display Output
	d. Definition of file records for query and report preparation.	
3.2.2.	File Maintenance	APPENDICES
	a. Modification of file control information	A. Sample Forms
	b. Update of file records	B. Computer Run Submission Procedures
	(1). Add records.	B.1. Update/Maintenance of Working-File
	(2). Modify records.	B.2. Retrieval/Reports against Working-File
	(3). Delete records.	

TABLE OF CONTENTS

## SECTION 1

### INTRODUCTION

#### 1.1. Purpose.

The general "working-file" database facility has been established to provide Naval Intelligence Command personnel with the capability to establish and maintain individual working files within the Naval Intelligence Command On-line System (NICOLS). Designed with an emphasis on rapid implementation and ease of use, the working-file facility combines database management capabilities with a

query and report writing facility known as CULPRIT.

This Users Guide is designed for use by NAVINTCOM personnel who have a need for immediate computer support. You should consult the NIPSSA Database Administration staff or your organization's ADP coordinator to be sure the working-file facility will support your application.

The facility and users guide is also available for use by non-NAVINTCOM personnel on an as-available basis. The purpose is to make a classified automated data processing capability accessible to persons who participate in the intelligence cycle but have no access to other classified processing environments.

## SECTION 2

### CAPABILITY DESCRIPTION

#### 2.1. Facility Capabilities.

The working-file database facility has been designed to:

1. Establish an unlimited number of individual files for one or more users.
2. Store up to 2000 records in each file in an order determined by the user.
3. Store up to 500 characters of user-defined data in each record within the file.

4. Maintain the following system information on each file for reference purposes:

- a. File security classification and handling.
  - b. File creation and update dates.
  - c. File user identification.
  - d. File titles.
5. Print the file contents or selected portions of the file in various orders and report formats.

## SECTION 3

### CREATING AND MAINTAINING THE WORKING-FILE

#### 3.1. Overview.

This portion of the Users Guide describes the procedures for creating and maintaining your working file. Each function is described separately using an illustration of the standard input form and instructions for completing the form. A "training file" is used throughout this Guide as an example to illustrate how to define your file and prepare your input forms. The instructions use some terms which may need clarification. The terms used throughout the Guide are:

Character - a letter, number, or special symbol such as a "+", "-", or "\*".

Data Field - a collection or combination of characters identifying a specific item of information, such as date or social security number.

Field Position - a specific space or group of spaces defined on an input form.

Record - a collection of related data fields that have been defined on the standard working-file input form

and which will be stored as a database record in the NICOLS system.

File - a collection of records belonging to a specific user. For example, a personnel file is made up of a collection of records on individuals within an organization.

Input Form - a form designed by the user which designates data field positions and which will be used for coding data.

#### 3.2. File Creation and Maintenance Functions.

The remainder of this section describes the various steps you should follow to create and maintain your individual working-file. Illustrations of the standard working-file input forms are provided using a training file as an example. Copies of all forms for reproduction use are provided in Appendix A.

File creation and maintenance steps are separated into three groups:

##### 1. File Creation.

- a. Identification of the file user to NICOLS.

- b. Identification of the file control information to NICOLS.
- c. Definition of file records for maintenance.
- d. Definition of file records for query and report preparation.

## 2. File Maintenance.

- a. Modification of file control information.
- b. Update of file records:
  - (1). Add records.
  - (2). Modify records.
  - (3). Delete records.

## 3. File Deletion.

The steps in group one are performed only one time when you initially define your file to the NICOLS system.

The steps in group two are performed any time you wish to update your file using the input forms you designed in group one.

The step in group three is performed when you no longer have need for your working-file and wish to delete the entire file from the NICOLS database.



## GROUP I - STEP 1

### IDENTIFICATION OF THE FILE USER TO NICOLS

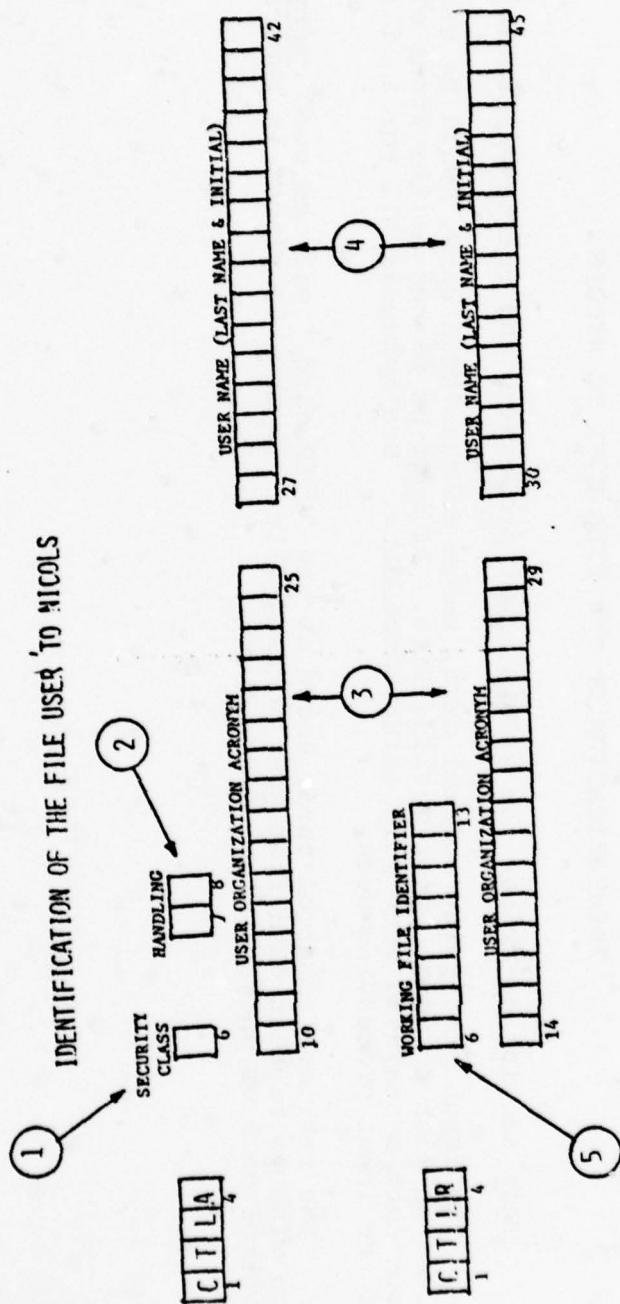
This section describes the NICOLS input processing and retrieval control cards.

The input processing control card, known as the "CTLA card," will be used every time you wish to update your working-file. It must be placed on the front of the card deck containing your file maintenance data. See Appendix B.1 for an illustration of the input processing deck.

The retrieval control card, known as the "CTLR card," will be used every time you wish to extract data from your working-file. It must be placed in the retrieval control deck as shown in Appendix B.2.

3.3(1.1)

IDENTIFICATION OF THE FILE USER TO NICOLS



## CTLA INPUT LINE

Item	Field Position On Input Form
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	11
12	12
13	13
14	14
15	15
16	16
17	17
18	18
19	19
20	20
21	21
22	22
23	23
24	24
25	25
26	26
27	27
28	28
29	29
30	30
31	31
32	32
33	33
34	34
35	35
36	36
37	37
38	38
39	39
40	40
41	41
42	42
43	43
44	44
45	45
46	46
47	47
48	48
49	49
50	50
51	51
52	52
53	53
54	54
55	55
56	56
57	57
58	58
59	59
60	60
61	61
62	62
63	63
64	64
65	65
66	66
67	67
68	68
69	69
70	70
71	71
72	72
73	73
74	74
75	75
76	76
77	77
78	78
79	79
80	80
81	81
82	82
83	83
84	84
85	85
86	86
87	87
88	88
89	89
90	90
91	91
92	92
93	93
94	94
95	95
96	96
97	97
98	98
99	99
100	100

Data Field Name and Description	
DATE	DATE OF BIRTH
NAME	FULL NAME
SEX	MALE OR FEMALE
AGE	CURRENT AGE IN YEARS
HEIGHT	HEIGHT IN METERS
WEIGHT	WEIGHT IN KILOGRAMS
BLOOD_PRESSURE	SYSTOLIC AND DIASTOLIC PRESSURE
HEART_RATE	BEATS PER MINUTE
GLUCOSE_LEVEL	BLOOD GLUCOSE LEVEL IN MG/DL
CHOLESTEROL_LEVEL	TOTAL CHOLESTEROL LEVEL IN MG/DL
HEMOGLOBIN_A1C	AVERAGE HEMOGLOBIN A1C LEVEL (%)
DIABETES_STATUS	TYPE 1, TYPE 2, OR PRE-DIABETES
INSULIN_USAGE	DAILY INSULIN DOSE IN UNITS
COMPLICATIONS	LIST OF DIABETES-RELATED COMPLICATIONS
PHYSICAL_ACTIVITY	DAILY PHYSICAL ACTIVITY LEVEL
NUTRITION_INTAKE	DAILY NUTRITIONAL INTAKE RECORD
MEDICATION_HISTORY	RECORD OF ALL MEDICATIONS TAKEN
LABORATORY_RESULTS	RECORD OF RECENT LABORATORY TEST RESULTS
DOCTOR_VISITS	RECORD OF VISITS TO THE ENDOCRINOLOGIST
PATIENT_COMPLIANCE	ASSESSMENT OF PATIENT COMPLIANCE WITH TREATMENT
QUALITY_OF_LIFE_SCORE	SCORE ON A QUALITY OF LIFE SCALE
PSYCHOLOGICAL_HEALTH	ASSESSMENT OF PSYCHOLOGICAL WELL-BEING
SOCIAL_SUPPORT_NETWORK	ASSESSMENT OF SOCIAL SUPPORT AVAILABLE
FINANCIAL_STABILITY	ASSESSMENT OF FINANCIAL STABILITY
WORKING_CONDITIONS	ASSESSMENT OF WORKING CONDITIONS
ENVIRONMENTAL_FACTORS	ASSESSMENT OF ENVIRONMENTAL FACTORS
GENETIC_TESTING_RESULTS	RESULTS FROM GENETIC TESTING FOR DIABETES PREDISPOSITION
PREVENTIVE_CARE_SCHEDULE	SCHEDULE FOR PREVENTIVE CARE VISITS
EMERGENCY_CONTACT_INFORMATION	CONTACT INFORMATION FOR EMERGENCY SERVICES
LEGAL_DOCUMENTS_ON_FILE	STATUS OF LEGAL DOCUMENTS ON FILE
TRANSPORTATION_OPTIONS	AVAILABLE TRANSPORTATION OPTIONS
HOUSING_SITUATION	CURRENT HOUSING SITUATION
HEALTH_INSURANCE_COVERAGE	DETAILS OF HEALTH INSURANCE COVERAGE
ACCESSIBILITY_TO_HEALTHCARE	ASSESSMENT OF ACCESSIBILITY TO HEALTHCARE FACILITIES
COMMUNITY_RESOURCES_AVAILABLE	LIST OF COMMUNITY RESOURCES AVAILABLE
PROVIDER_COORDINATION_EFFECTIVENESS	EFFECTIVENESS OF PROVIDER COORDINATION
PATIENT_education_resources_available	AVAILABLE EDUCATIONAL RESOURCES FOR PATIENTS
TELEHEALTH_SERVICES_AVAILABLE	TELEHEALTH SERVICES AVAILABLE
REMOTE_MONITORING_DEVICES_AVAILABLE	AVAILABLE REMOTE MONITORING DEVICES
DIETARY_COUNSELING_AVAILABLE	DIETARY COUNSELING SERVICES AVAILABLE
SMOKING_CESSATION_PROGRAM_AVAILABLE	SMOKING CESSATION PROGRAM AVAILABLE
ALCOHOL_USE_COUNSELING_AVAILABLE	ALCOHOL USE COUNSELING SERVICES AVAILABLE
MENTAL_HEALTH_SUPPORT_AVAILABLE	MENTAL HEALTH SUPPORT SERVICES AVAILABLE
TRANS�ATELATION_SERVICES_AVAILABLE	LANGUAGE TRANS�ATELATION SERVICES AVAILABLE
ADDITIONAL_NOTES	ANY OTHER RELEVANT INFORMATION

1 SECURITY CLASSIFICATION OF FILE. Enter the security  
6 classification for the file. Valid codes are:

U - Unclassified      O - For Official Use Only  
C - Confidential      S - Secret  
T - Top Secret

2 7 - 8 SECURITY RELEASIBILITY/HANDLING CODE. Enter the releasability/handling code for the file. This code must conform to DIAM 65-19, "Standard Security Markings." No releasability restriction is imposed if the field is left blank.

3 10-25

USER ORGANIZATION ACRONYM. Enter the acronym of your organization. Check with the NIPSSA Database Administration staff or your organization ADP coordinator for the proper acronym spelling.

4 27-42

USER PERSON NAME. Enter your name. Start your last name in the leftmost position of the field. Follow your last name with your initials. Do not use periods or commas. NOTE: This is your permanent identifier to the NICOLS system. Be sure to spell it the same each time.

C'TLR INPUT LINE

5 6-13 WORKING FILE IDENTIFIER. Enter the name of the working file as assigned.

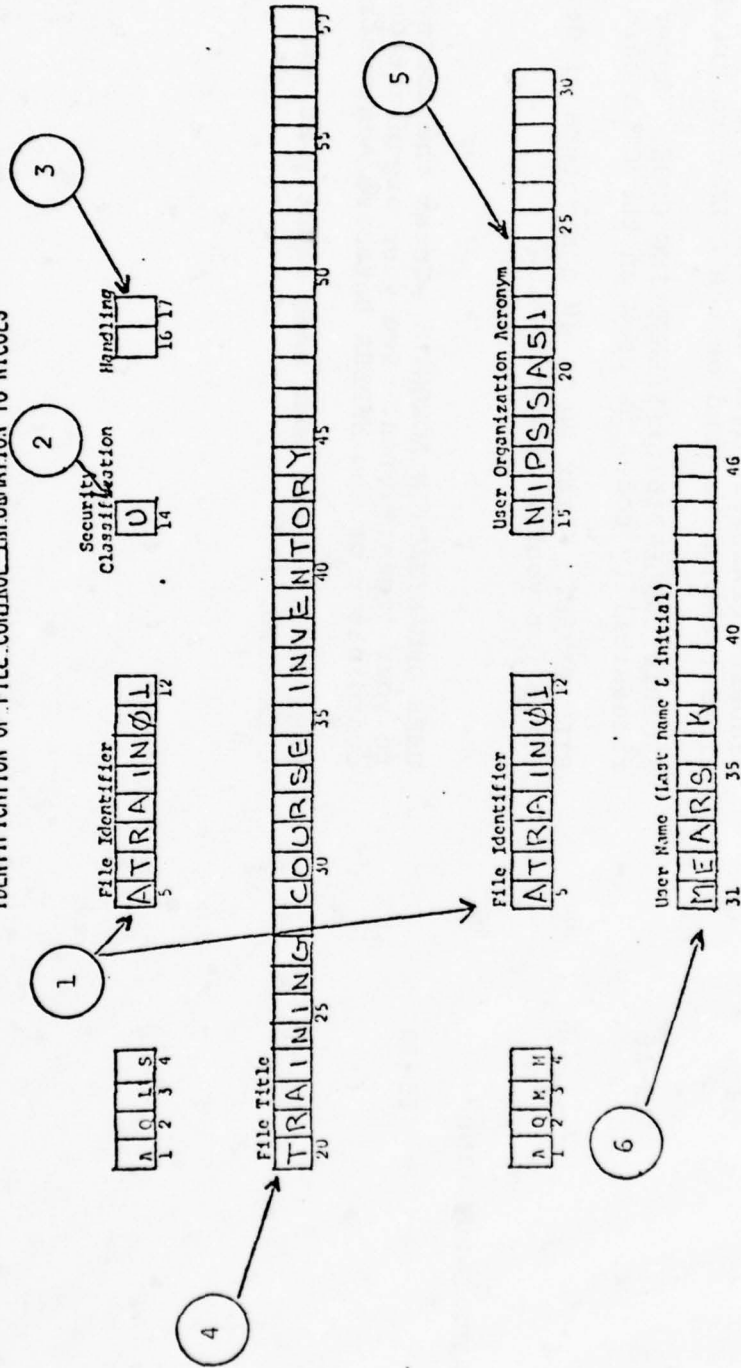
### 3.5 IDENTIFICATION OF THE FILE USER TO NICOLS

## GROUP I STEP 2

### IDENTIFICATION OF THE FILE CONTROL INFORMATION TO NICOLS

This section defines the steps necessary to define your working-file to the NICOLS system. When you have completed this section, you will have prepared the necessary computer input to create a working-file control record. This record identifies your working-file and provides the control information which permits you to store file records in the database.

# IDENTIFICATION OF FILE CONTROL INFORMATION TO NICOLS





# IDENTIFICATION OF THE FILE CONTROL INFORMATION TO NICOLS

## AQLS INPUT LINE

Item	Field Position on Input Form	Data Field Name and Description
1	5-12	FILE IDENTIFIER. Obtain an identifier for your file from your organization ADP coordinator or the NIPSSA Database Administration staff. Enter the identifier in the space provided on the form.
2	14	SECURITY CLASSIFICATION OF FILE. Enter the same classification as used on the CTLA card previously.
3	15-16	SECURITY RELEASIBILITY/HANDLING CODE. Enter the same releasibility codes as used on the CTLA card.
4	20-59	FILE TITLE. Enter up to 40 characters of descriptive title for your working-file.

## AQMM INPUT LINE

5	15-30	USER ORGANIZATION ACRONYM. Enter the acronym assigned to your organization. See your organization ADP coordinator or the NIPSSA Database Administration staff.
6	31-46	USER NAME. Enter your name exactly as entered on the CTLA card.

# IDENTIFICATION OF THE FILE CONTROL INFORMATION TO NICOLS

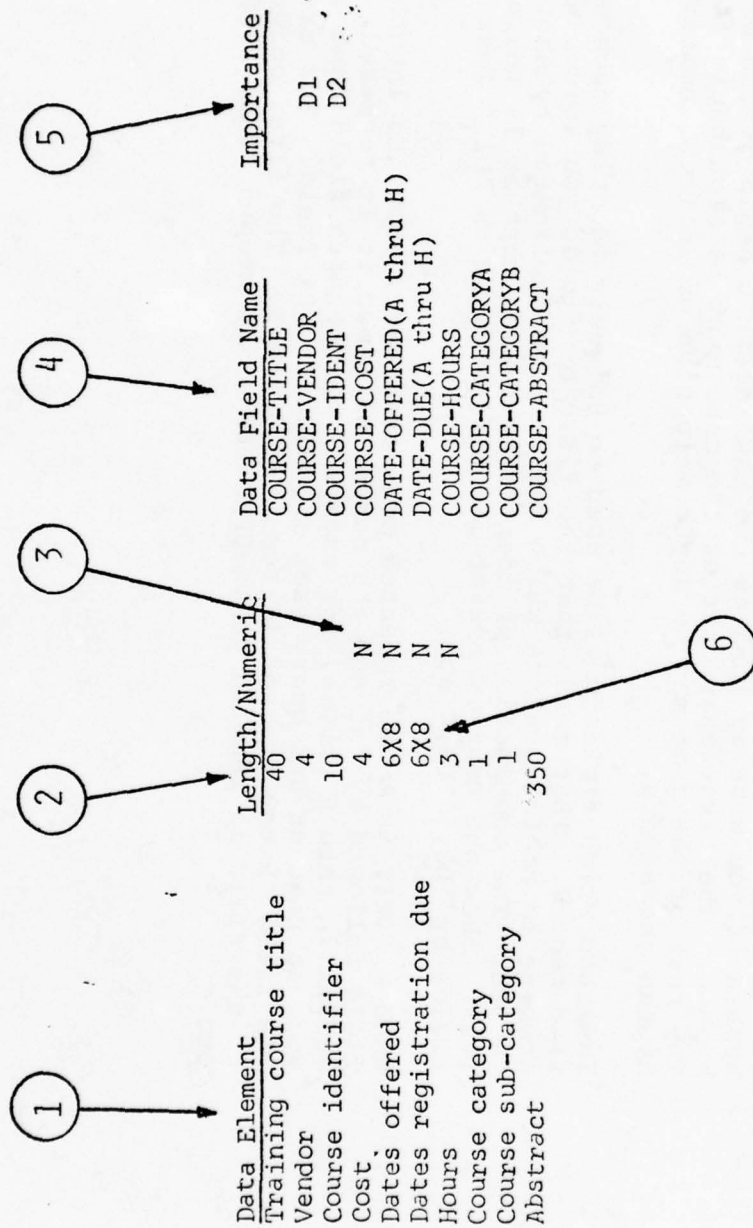
### GROUP I STEP 3

#### DEFINITION OF FILE RECORDS FOR MAINTENANCE (PART 1)

This section describes the procedures used to define the way the working-file detail record will look. When you have completed both parts of this section, you will have created an entry form which can be reproduced and used to prepare information for loading your working-file.

Part 1 assists you in determining what data fields you wish to store in your working-file, how to define their characteristics, assign names by which the NICOLS system will recognize each data field, and determine the order in which file records will be stored in the database.

3.12(1.1)



### DEFINITION OF FILE RECORDS FOR MAINTENANCE (PART 1)

- 1 Write down a list of data fields you wish to store in the working-file.
- 2 Indicate the number of characters (length) of each data field.
- 3 Mark "N" beside those elements which contain only numbers.
- 4 Write the name you wish to use when referring to each data field. This name may be up to 16 characters long. You may join two words with a hyphen if you wish as long as the name doesn't begin or end with a hyphen. The first character of the name must be alphabetic (A-Z). The rest of the name may be any combination of letters, numbers, or hyphen characters.
- 5 Indicate which fields will be used to uniquely identify each working-file record. This means that the field or fields you select will combine to contain values which will not be duplicated by another record. The combination of identifier fields must be 16 characters or less. Show the most important identifier field as "D1," less important fields by "D2," "D3," etc.
- 6 When a field is to be repeated more than once, show the length of the field followed by "x" and the number of times it is repeated. Put a suffix (A thru H, above, for example) on the data field name so that you can request and query each individual data field. For example, the field "Date-Offered" is repeated 8 times. The first occurrence of the field is labeled "DATE-OFFEREDA", the second "DATE-OFFEREDB", etc.

#### DEFINITION OF FILE RECORDS FOR MAINTENANCE (PART 2)

Part 2 describes how to prepare your input data entry form. This form, once completed, may be reproduced and used to actually code information to be stored in the working-file.



Record Identifier

TRAINING

7 9 11  
(Duplicate columns)

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

VENTON, COURSE IVENT  
13 17 19 21 23 25 27  
each input line)

**Note:** An asterisk (\*) in any position of the input form will cause the corresponding position in the working-file record to be blanked out. Any other character in any position of the input form will be stored in the corresponding position in the working-file record, replacing the character that previously existed in that position in the record.

[illegible]

DEFINITION OF FILE RECORDS FOR MAINTENANCE (PART 2)

- 1 Enter a "S" to instruct the system to store file records.
- 2 Enter the assigned name of the working-file.
- 3 Place vertical lines separating individual data fields which are used to uniquely identify each working-file entry. These are the fields which you marked "D1," "D2," etc., in part 1.
- 4 Write the name of each field in the record identifier below the spaces. These are the data fields which you have determined will uniquely identify each record in your working-file.
- 5 Mark out unused positions to reduce errors when entering data values in the identifier fields.
- 6 Place vertical lines separating individual data fields in the main area of the working-file record. Each space contains a single character of data. Allow as many spaces as required for the maximum number of characters in a data field.
- 7 Write the name of each field in the main area of the working-file record form above the spaces which will contain the field's data.
- 8 Mark out unused positions to reduce errors when entering data in the main area of the entry.

#### GROUP I - STEP 4

##### DEFINITION OF FILE RECORDS FOR QUERY AND REPORT PREPARATION

This section describes how to prepare instructions to the NICOLS query facility so that you can create your own reports from the working-file. When you have completed this section, the query facility will be able to locate each data field in your file records.

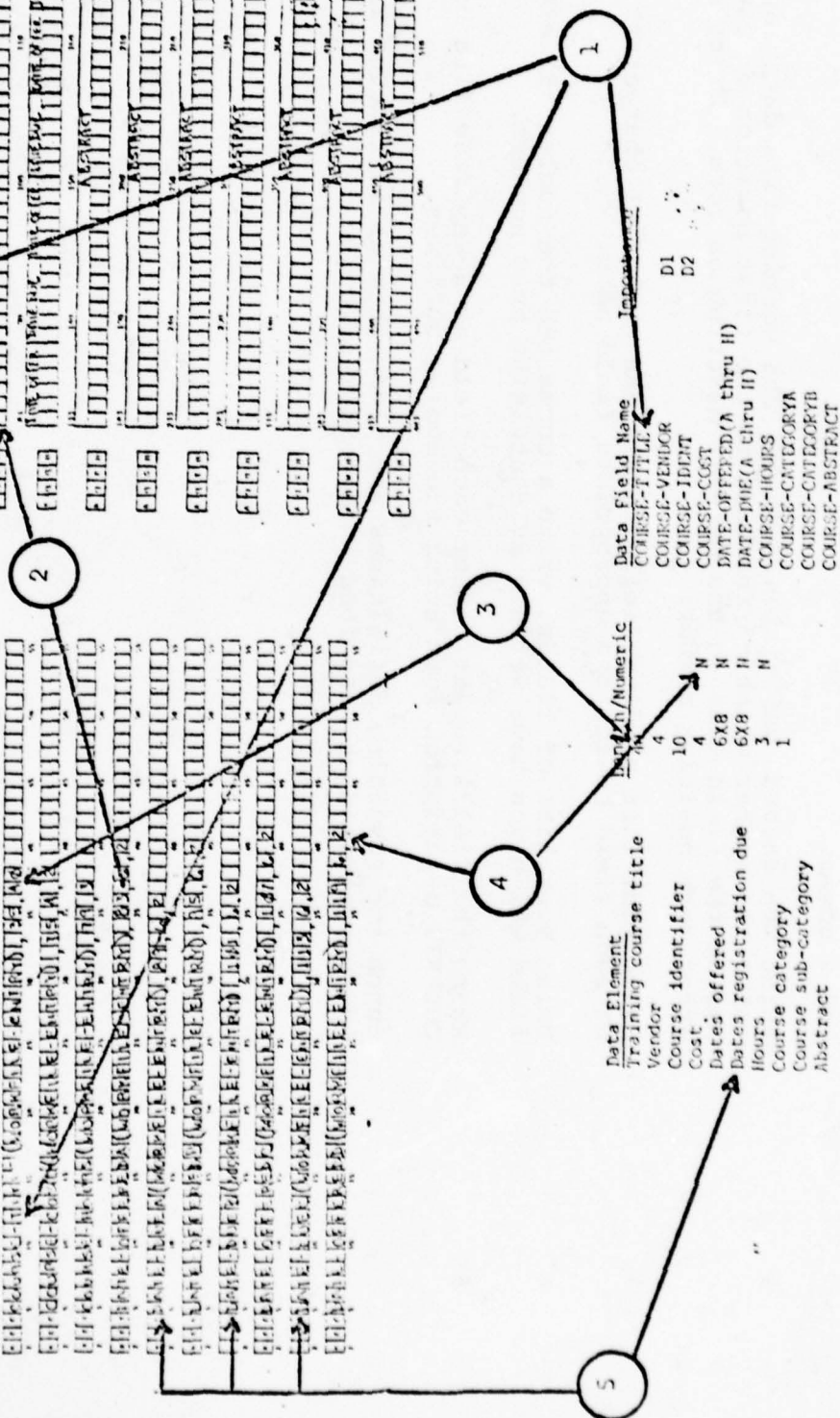
3.19

##### DEFINITION OF FILE RECORDS FOR QUERY AND REPORT PREPARATION

[illegible]

THE UNIVERSITY OF CHICAGO

A photograph of a document with a repeating pattern of the word "COPIES" and a large diagonal line drawn across it. The document appears to be a form or a ledger with multiple columns and rows. The word "COPIES" is printed in a bold, sans-serif font and is repeated across the top of the document. A large, thick, black diagonal line is drawn across the entire page, from the top left to the bottom right, crossing out the content. The background is a light, textured surface, possibly a wall or a backdrop.





## DEFINITION OF FILE RECORDS FOR QUERY AND REPORT PREPARATION

Using the RECORD DEFINITION FORM, mark the data fields to exactly match the data input form you prepared in Group I, Step 3.

- 1 Write the name of the data fields, one to a line, on the CULPRIT Record Description form. Follow each name with "WORKFILE-ENTRY" and a comma.
- 2 Using the Record Definition Form and the working-file data input form, enter the number which corresponds to the first position of each field beside the field name on the CULPRIT description form. Write a comma after each position number.
- 3 Using your list of data fields, enter the number of characters (length) of each field beside the appropriate field name.
- 4 Using your list of fields, write a comma and the number "2" after any field which you have defined as containing only numbers.
- 5 Where the fields repeat, enter each field as a separate line on the CULPRIT description form using succeeding suffixes.  
  
Punch the resulting definitions and place them as instructed into the query control deck described in Appendix B.2.

3.21 (1.1)

DEFINITION OF FILE RECORDS FOR QUERY AND REPORT PREPARATION



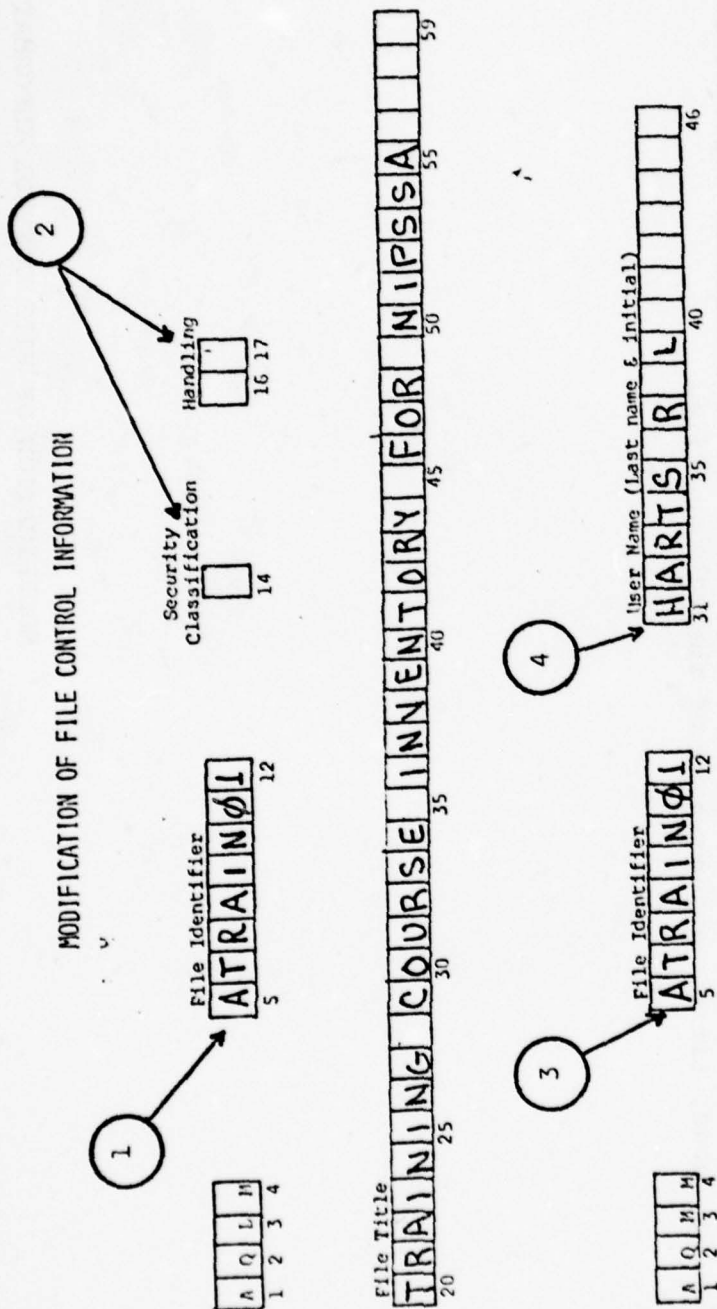
GROUP II STEP 1

MODIFICATION OF FILE CONTROL INFORMATION

This section describes the steps necessary to modify the security classification, releasability code, title, or user name of the working-file.

MODIFICATION OF FILE CONTROL INFORMATION

# MODIFICATION OF FILE CONTROL INFORMATION



# MODIFICATION OF FILE CONTROL INFORMATION

## AQLM INPUT LINE

Item	Field Position on Input Form	Data Field Name and Description
1	5-12	FILE IDENTIFIER. Enter the assigned working-file identifier.
2		Enter the field to be changed.
AQMM INPUT LINE		
3	5-12	FILE IDENTIFIER. Enter the assigned working-file identifier.
4	31-46	USER NAME. Enter the changed person name.

NOTE: The user organization information cannot be changed except by the Database Administration staff.

It is only necessary to prepare the input lines above when the data field on that line changes. The AQLM is not required, for example, if the only change is the name of the person controlling the working-file.

GROUP II STEP 2

ADD RECORDS

This section describes the steps necessary to add data records to the working-file you have created.





#### ADD RECORDS

- 1 Enter a "S". This instructs the NICOLS system to store a new record in your working-file.
- 2 Enter the assigned working-file identifier. This identifier will be duplicated into all punched cards produced for this file.
- 3 Enter the data values which will uniquely identify the file record for later referral. These fields will be duplicated into all punched cards produced for this file record.
- 4 Enter data in the desired fields. Data for alphabetic fields normally begins in the left-most position of the field. Numeric data normally ends in the right-most position of the field.
- 5 It is advisable to enter zeros into all numeric fields which are not initially filled with data. This makes sure that a numeric field contains a numeric value and prevents errors in processing.
- 6 Numeric fields whose data is not large enough to fill all available positions should be filled with zeros from the left-most position of the field to the beginning of the numeric data.
- 7 Cross through entry lines which are not used. This assists the person who punches the lines in deciding which lines to punch and which to ignore.

The punched cards which are created as a result of this section are inserted into the computer control deck as described in Appendix B.1.

#### MODIFY RECORDS

This section describes the steps used when changing the information you have previously stored in the working-file.



## MODIFY RECORDS

- 1 Enter "M." This instructs the NICOLS system to modify a working-file record.
- 2 Enter the assigned working-file identifier.
- 3 Enter the data which uniquely identifies the record to be modified.
- 4 Alphabetic data may be modified by simply replacing the characters or by entering an asterisk to delete the previous character in that position.
- 5 Numeric data may be modified by replacing the digits or by entering zeros to delete the data in the field and set its value to zero.
- 6 Cross out unused entry lines to assist the person who punches the cards in determining which lines to punch and which to ignore.

The punched cards which are created as a result of this section are inserted into the computer control deck as described in Appendix B.1.

## DELETE RECORDS

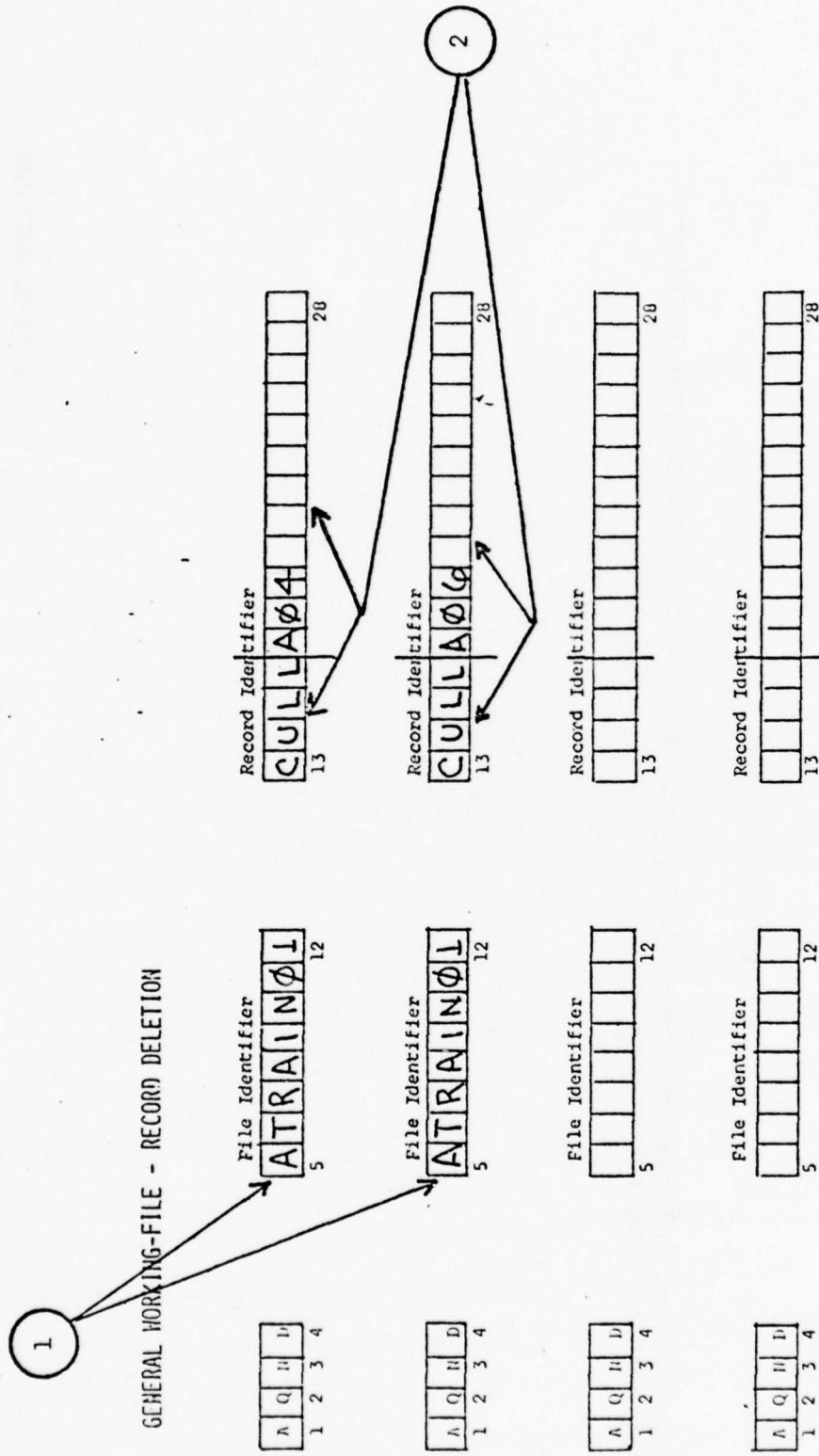
This section defines the steps required to delete a record from your working-file which is no longer required.

3.35

DELETE RECORDS

PRECEDING PAGE BLANK





DELETE RECORDS

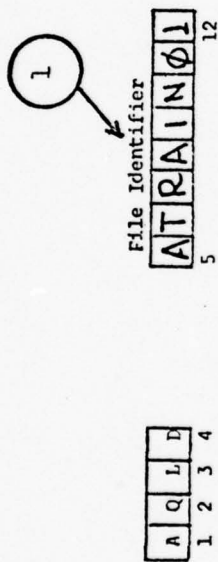
- 1 FILE IDENTIFIER. Enter the assigned identifier for the working-file.
- 2 RECORD IDENTIFIER. Enter the values which uniquely identifies the record to be deleted.

The punched card which results from this section is inserted into the computer control deck as described in Appendix B.1.

#### FILE DELETION

This section defines the steps required to delete a working-file from the data-base when it is no longer required. Once the working-file has been deleted, all data within the file is physically erased and no longer accessible.

## FILE DELETION FROM NICOLS



CAUTION: This will erase your entire Working-File from the database.

FILE DELETION

1 FILE IDENTIFIER. Enter the assigned identifier of the working-file.

The punched card which results from this section is inserted into the computer control deck described in Appendix B.1.



## SECTION 4

### RETRIEVING DATA FROM THE WORKING-FILE

#### 4.1. Overview.

This portion of the Users Guide describes the procedures for creating reports to display data stored in your working-file. Each output function is described separately, using illustrations to assist you. Retrieval support is provided through the CULPRIT report and query facility. This facility provides a very fast and easy means to producing useful reports.

This Guide will not describe all of the features of CULPRIT. Instead, it will concentrate on showing you how to prepare basic reports and queries. Once you have mastered the basic features of the facility, the Database Administration staff or your organization ADP coordinator will assist you in preparation of more

complex reports. A copy of the CULPRIT User's Manual, published by the package vendor, is available from the Database Administration staff upon request.

#### 4.2. Retrieval Support.

The working-file system provides you with two retrieval aids:

- a. A prepared CULPRIT report which prints the contents of your working-file in a predefined format. This report, called the "file list", is particularly useful when updating the contents of your file. See Appendix B.2.
- b. Detailed instructions for preparing your own reports. Each parameter used by CULPRIT is described in detail.

4.1(1.1)

PRECEDING PAGE BLANK

4. RETRIEVING DATA FROM THE WORKING-FILE

## PREPARING YOUR OWN REPORTS

### DEFINING THE REPORT CONTROL INFORMATION

This section describes the first step which must be taken to define a report to the CULPRIT facility. When you have completed this step, you will have created a title for the report and determined the order in which the data will be displayed.



DEFINING THE REPORT CONTROL INFORMATION

- 1 REPORT NUMBER. Enter a two-digit report identifier ranging from "02" through "99". Each of your reports should have a different identifier so that you can use several reports at the same time.
- 2 LINES PER PAGE. Enter a two-digit number which defines the number of report lines to print on a page. The maximum number is 58. To compute the number of available lines, subtract the number of lines in your report heading (including blank lines) from 58.
- 3 TOTAL/DETAIL FLAG. This position permits you to select whether you wish the report to automatically total numeric fields, print only the basic report lines, or both. The default value (leave the field blank) is both. Valid codes are:
  - D - Detail basic lines only
  - T - Total lines only (useful for summarizing groups of data)
- 4 REPORT SORT FIELDS. This entry area (and that on the following line) are used to define the order in which you wish your data to be printed. Enter the field name of the principal field which determines the output order.
- 5 You can instruct the system to do extra spacing or skip to a new page if the value in the principal field changes. Enter a comma followed by one of these codes:
  - 1 - Skip to a new page
  - 0 - One extra space after the last detail line of the previous field value
  - - Two extra spaces after the last detail line of the previous field value
- 6 You can name as many fields in the sort order as you desire. Separate them with commas. If there are too many sort fields for one line, begin a new field name on the continuation line.

# CULPRIT SORT AND TITLE STATEMENTS

Report Number **42** (Duplicate columns 2-3 for each line coded)

Report Sort Fields

2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
D		C		O		U		R		S		E		-		V		E		N		D		O		R		,		1		,		C		O		U		R		S		E		-		I		D		E		N		T		

Report Sort Fields continued

16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

Report Sort Fields continued

3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60																				
A		L		L		C		O		U		R		S		E		S		B		Y		V		E		N		D		O		R		C		O		S		T		I		N		G		L		E		S		S		T		H		A		N		I		I		D		O		P		H	



DEFINING THE REPORT CONTROL INFORMATION (CONTINUED)

7

Enter the title you wish printed at the top of each page of the report. The CULPRIT facility automatically centers the title. A page number and report date are automatically added to the title line.

The punched cards resulting from this section are added to those prepared in the two sections following to form a report definition. This deck is inserted as shown in Appendix B.2 to produce the report.

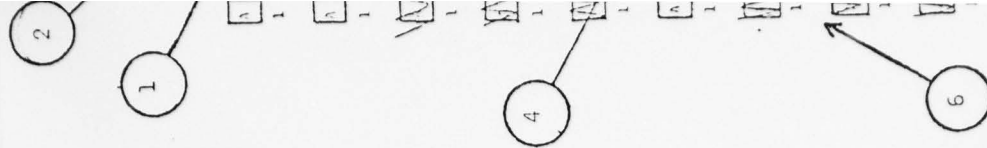
#### DEFINING THE REPORT HEADING

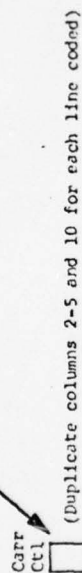
This section describes how to define a heading for your report. CULPRIT permits the user to define up to eight separate lines of heading (not including any blank lines you request). The working-file facility reserves heading line one for printing security classification. This is described in a later section.

When you have completed this section, you will have defined the basic heading lines for your report.

PRECEDING PAGE BLANK

DEFINING THE REPORT HEADING



[illegible]

#### DEFINING THE REPORT HEADING

1 REPORT NUMBER. Enter the number of the report you are preparing, ranging from "02" to "99".

2 LINE TYPE. Enter "4" to identify a heading line.

3 LINE NUMBER. Enter "2" through "8" as desired, identifying the heading line to be defined.

4 CARRIAGE CONTROL. You may define controls which force extra blank lines or a skip to a new page. The requested function occurs before the line is printed. Valid codes are:

- 1 - Skip to a new page
- 0 - Leave one blank line before the line to be printed
- - Leave two blank lines before the line to be printed

The default (leaving the position blank) is normal line to line spacing with no extra blank lines.

NOTE: All of the descriptions on this input description page refer to one line of the report. A new form should be used for each line. Once you have become familiar with CULPRIT, a general description form, supplied by the CULPRIT vendor, may be used.

5 LINE POSITION. The printer used by the NICOLS computer is capable of printing lines 132 characters long. Enter the desired print location for the left-most position of the information associated with the field name (see 7 below) or the literal text (see 6 below).

6 LINE CONTENTS. You may enter either text (literal) enclosed by single quote marks or

7 the name of a field which you have used in the sort control (previous section).

#### DEFINING THE REPORT BODY

This section describes the steps necessary to tell CULPRIT how to print the main body of your report. You may have noticed that the report heading did not include any description about the contents of the report. A special feature of CULPRIT permits you to describe the data you wish printed and the headings for each data field. CULPRIT then automatically distributes the data across the report page, inserts the headings, and centers the data under the headings.

When you have completed this section, you will have defined the entire report display description.

DEFINING THE REPORT BODY



CULPRIT REPORT DEFINITION FORM

1

CULPRIT REPORT DEFINITION FORM

2

3

4

5

6

7

8

9

10

11

Report Number

Line Type

Line Number

Line Contents

Line Position

Carr Ctl

(Duplicate columns 2-5 and 10 for each line coded)

Line Position	Line Contents	Line Number	Line Type	Report Number
1	COURSE-IDENT	1	5	2
2	HH			
3	'COURSE'			
4	'IDENT'			
5	'TITL'			
6	'E'			
7				
8				
9				
10				
11				

1

Line Position	Line Contents	Line Number	Line Type	Report Number
1	COURSE-TITLE	2		
2	HH			
3	'COURSE'			
4	'TITLE'			
5				
6				
7				
8				
9				
10				
11				

1

Line Position	Line Contents	Line Number	Line Type	Report Number
1	COURSE-COST	3		
2	DP			
3	HH			
4	'COST'			
5	'OF'			
6	'COURSE'			
7				
8				
9				
10				
11				

1

Line Position	Line Contents	Line Number	Line Type	Report Number
1	COURSE-HOURS	4		
2	DP			
3	HH			
4	'CLASS'			
5	'HOURS'			
6				
7				
8				
9				
10				
11				

1

Line Position	Line Contents	Line Number	Line Type	Report Number
1	DATE-OFFERED	5		
2	ED			
3	HH			
4	'NEXT DATE'			
5	'OFFERED'			
6				
7				
8				
9				
10				
11				

1

Line Position	Line Contents	Line Number	Line Type	Report Number
1	DATE-DUE	6		
2	ED			
3	HH			
4	'NOTIFICATIONS'			
5	'DUE BY'			
6				
7				
8				
9				
10				
11				

#### DEFINING THE REPORT BODY

1 REPORT NUMBER. Enter the number of the report you are preparing, ranging from "02" through "99".

2 LINE TYPE. Enter "5" to identify a report body line.

3 LINE NUMBER. Enter "1" through "8" as desired, identifying the report body line. CULPRIT prints report body lines in line number order.

4 CARRIAGE CONTROL. You may define controls which force extra blank lines or a skip to a new page. The requested function occurs before the line is printed. Valid codes are:

- 1 - Skip to a new page
- Ø - Leave one blank line before the line to be printed
- - Leave two blank lines before the line to be printed

The default (leaving the position blank) is normal line to line spacing with no extra blank lines.

NOTE: All of the descriptions on this input description page refer to one line of the report. A new form should be used for each line.

5 LINE POSITION. Enter an asterisk in the left-most position of the field. This instructs CULPRIT to generate automatic headings for the data field. Enter a number in the right-most positions of the field to tell CULPRIT what order to place the fields on the report page, progressing from left to right.

6 Enter the name of the data field you wish displayed.

7 Enter "HH" after the data field name, separated by one space. This defines the heading control to CULPRIT.

8 Enter a literal you wish to use as a heading for the column where the element will be displayed. Two literals will produce two heading lines.

DEFINING THE REPORT BODY



DEFINING THE REPORT BODY (CONTINUED)

- 9 Dates may be automatically formatted by CULPRIT. Enter "FD" after the name of a date field and CULPRIT will insert slashes (/) between the month, day, and year. The date field must be defined as a 6-digit field and contain only numbers or an error will result.
- 10 CULPRIT assumes all numeric fields contain financial (dollar) data. It will automatically insert a period between the second and third digits of a numeric field when displaying the field unless you tell it otherwise. For numeric fields with no decimal positions, enter "D0". If the numeric field has one decimal position, enter "D1"; enter "D3" for three positions, etc. The default is two decimal positions.
- 11 CULPRIT automatically formats all numeric fields. It will place commas where appropriate to separate thousands and millions. It will also display leading zeros if desired. Valid codes are:
- F0 - Leading zeros are suppressed; commas are not inserted.
  - F1 - n may = 1-9; Leading zeros are suppressed except for the last n digits; commas are inserted.
  - FN - No editing. The field is printed as it exists in the data field.
  - F\$ - Leading zeros are suppressed; commas are inserted; a floating dollar sign is inserted.
  - FS - Field is edited as a social security number.



#### DEFINING THE DATA SELECTION (QUERY) STATEMENTS

This section is optional with every report. If you omit the query statements from the report, CULPRIT will select and print desired data from every record in the working-file.

CULPRIT permits you to define a wide variety of query control statements. The primary statement is a test which determines whether the record you are reading is to be printed or not. This single statement type is adequate for a majority of queries you will be creating against your working-file. Should you have a need to prepare more complex reports or queries, you should contact your organization ADP coordinator or the Database Administration staff.



**CULPRIT QUERY STATEMENTS**

1 (Report Number) → 2 (Sequence Number) → 3 (Process Instructions) → 4 (Duplicate columns 2-4 on each card punched) → 5 → 6

Sequence Number: 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

Process Instructions: COURSE - COST H 100 DROP

[illegible]

5	6	7	8	9	10	15	20	25	30	35	40	45	50	55	60
---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----

[illegible][illegible][illegible][illegible]

5	1	5	10	15	20	25	30	35	40	45	50	55	60
---	---	---	----	----	----	----	----	----	----	----	----	----	----

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	-----

#### DEFINING THE DATA SELECTION (QUERY) STATEMENTS

- 1 REPORT NUMBER. Enter the number of the report you are preparing, ranging from "02" through "99".
- 2 SEQUENCE NUMBER. This field tells CULPRIT in what order query statements are to be executed. CULPRIT automatically sorts the query statements on this field so you can write statements and later insert another by using a sequence number between the two where the new statement is to appear. Sequence numbers "010" through "899" are available for your use. It is recommended that you add ten to each sequence number (10, 20, 30, etc.)
- 3 Enter the data name of the field you wish to test.
- 4 Enter the code for the type of test you wish made against the field.  
Valid codes are:
  - E - The field is equal to the test value
  - N - The field is not equal to the test value
  - L - The field is less than the test value
  - H - The field is greater than the test value.
- 5 TEST VALUE. Enter the value which is used to test the selected data field. If the data field is numeric, the test value may be written as illustrated. If the data field is alphanumeric, the test value must be enclosed in single quote (') marks.
- 6 RESULT FUNCTION. This field tells CULPRIT what to do if the result of the test is true. Valid codes are:
  - DROP - Discard the record. It will not be printed.
  - TAKE - Accept the record and print it.



## SECTION 6

### TELEPROCESSING OUTPUT

#### 6.1. Overview.

This portion of the Users Guide describes the procedure to generate CULPRIT reports from the working-file and store the output for later retrieval and display on the CRT Terminals. This section will be divided into two parts. Part I will explain how to generate the reports to be used for Teleprocessing Output and Part II will explain how to display the output on the CRT Terminals.

The Teleprocessing Output will be limited to 79 characters to a line, 20 lines to a page including any header, total and blank lines and will display only the first ten pages of any report generated on the CRT Terminal. Generated CULPRIT reports will be automatically deleted after 10 days.

#### 6.2. Part I - Generate CULPRIT reports from working-file.

To generate your CULPRIT reports for Teleprocessing Output, a

prior knowledge of how to create, maintain and retrieve data from the working-file in Sections 3 and 4 is strongly suggested. The same computer control deck illustrated in APPENDIX B.2 can be used to generate CULPRIT reports for Teleprocessing Output by making three card changes:

- a. In defining the report control information on the "2J" card (pg. 4.4), there are two changes to be made: 1) to the lines per page filed; and 2) add an output record size field.
- b. The second card of your computer control deck will have a different STEP EXEC card.
- c. A new card will be added after the STEP EXEC card called the PARAM card.

6.1(2.0)

TELEPROCESSING OUTPUT

CHANGE 1

DEFINING THE REPORT CONTROL INFORMATION

This section describes the two changes needed on the "2J" card of your computer control deck. Show special attention to steps 2 and 8. The remaining steps used are identical to that on pgs. 4.4 to 4.7.

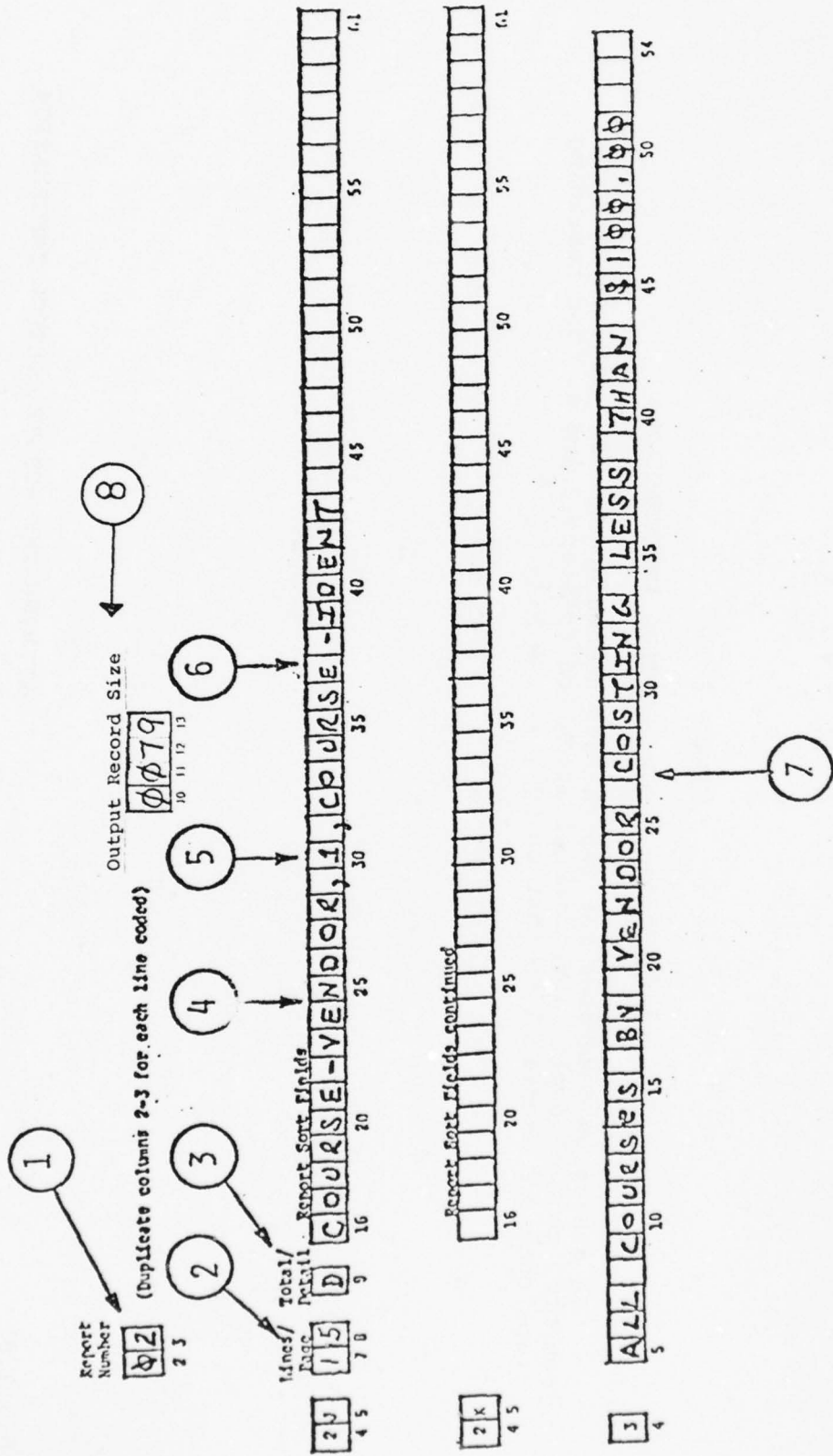
DEFINING THE REPORT CONTROL INFORMATION

6.3(2.0)

PRECEDING PAGE BLANK



# CULPRIT SORT AND TITLE STATEMENTS



DEFINING THE REPORT CONTROL INFORMATION

## DEFINING THE REPORT CONTROL INFORMATION

1      **REPORT NUMBER.** Enter a two-digit report identifier ranging from "02" through "99". Each of your reports should have a different identifier so that you can use several reports at the same time.

2      **LINES PER PAGE.** Enter in a two-digit number which defines the number of report lines to print on a page. The maximum number is 20. To compute the number of available lines, subtract the number of lines in your report heading (including blank lines) and any total lines in your report from 20.

3      **TOTAL/DETAIL FLAG.** This position permits you to select whether you wish the report to automatically total numeric fields, print only the basic report lines, or both. The default value (leave the field blank) is both. Valid codes are:

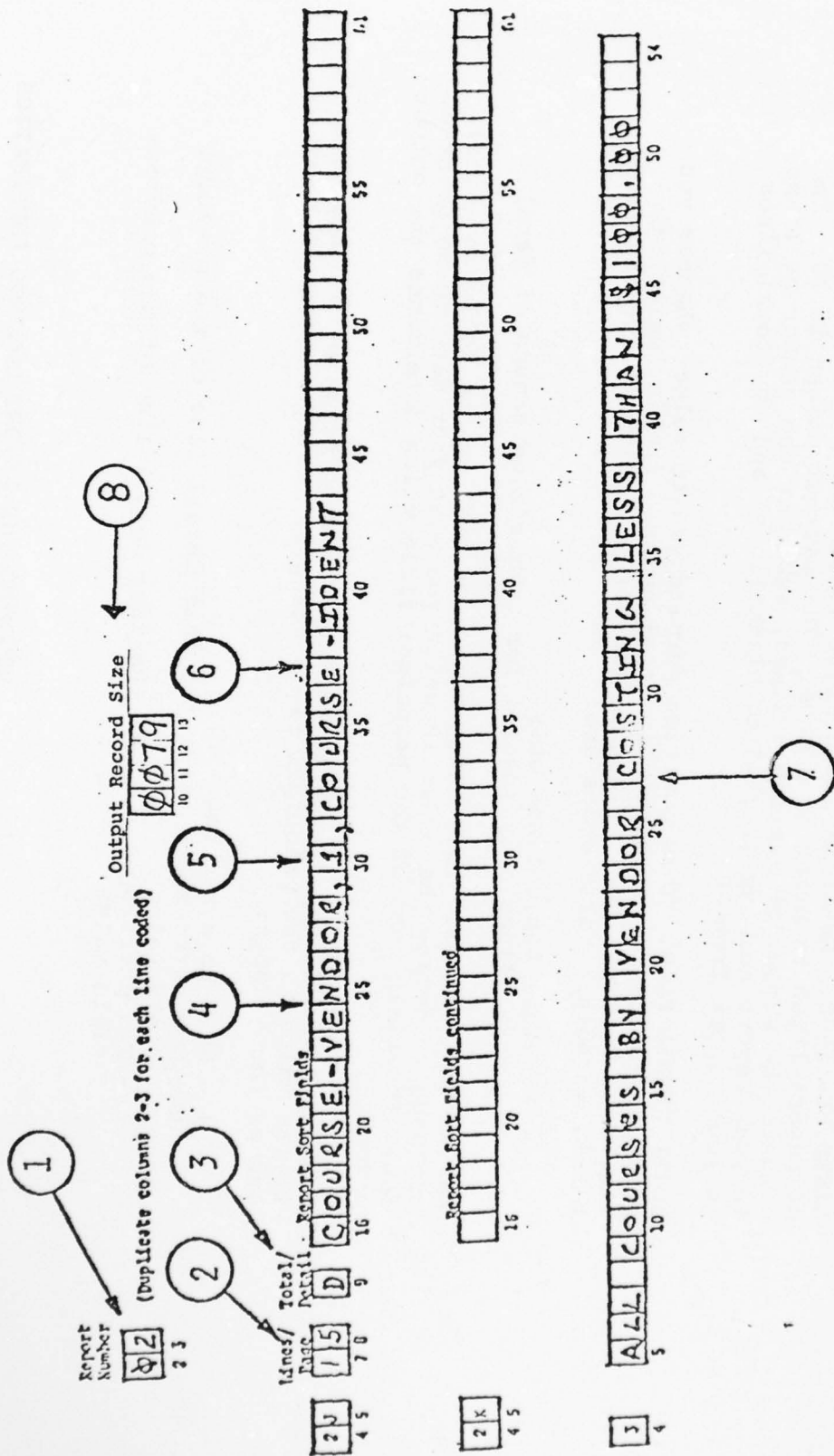
- D - Detail basic lines only
- T - Total lines only (useful for summarizing groups of data)

4      **REPORT SORT FIELDS.** This entry area (and that on the following line) are used to define the order in which you wish your data to be printed. Enter the field name of the principal field which determines the output order.

5      You can instruct the system to do extra spacing or skip to a new page if the value in the principal field changes. Enter a comma followed by one of these codes:

- 1 - Skip to a new page
- 0 - One extra space after the last detail line of the previous field value
- - Two extra spaces after the last detail line of the previous field value

# CULPRIT SORT AND TITLE STATEMENTS



DEFINING THE REPORT CONTROL INFORMATION

DEFINING THE REPORT CONTROL INFORMATION (CONTINUED)

- 6 You can name as many fields in the sort order as you desire. Separate them with commas. If there are too many sort fields for one line, begin a new field name on the continuation line.
- 7 Enter the title you wish printed at the top of each page of the report. The CULPRIT facility automatically centers the title. A page number and report date are automatically added to the title line.
- 8 OUTPUT RECORD SIZE. Enter "0079" to define the maximum record length of your report. This means that all field lengths and spaces between fields must not exceed 79 positions.

The punched cards resulting from this section are added to those prepared in the two sections following to form a report definition. This deck is inserted as shown in Appendix B.2 to produce the report.

## CHANGE 2

### CHANGE COMPUTER CONTROL DECK

This section describes what changes need to be made to your computer control deck. It will first describe and explain the PARAM card and second, it will show where the two changes take place in your computer control deck.

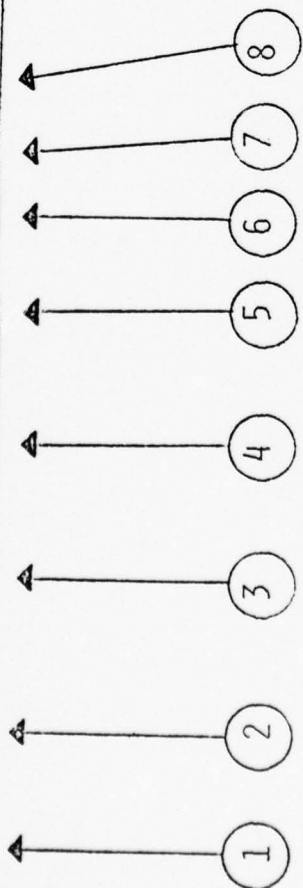
6.9(2.0)

PRECEDING PAGE BLANK

CHANGE COMPUTER CONTROL DECK



1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80



CHANGE COMPUTER CONTROL DECK

Every CULPRIT retrieval must have a PARAM statement. The PARAM statement must be preceeded and concluded with three single quotes; it also must have five fields and appear in the order given on previous page, each followed by a comma.

1 Two slashes are entered in card columns 1 and 2.

2 Enter PARAM="" in card columns 4 thru 12.

3 Replace "AAAAA" with the user's organization acronym. This field can have up to 16 characters and must be compatible with item 3 on pg. 3.5. This field starts in card column 13 followed by a comma.

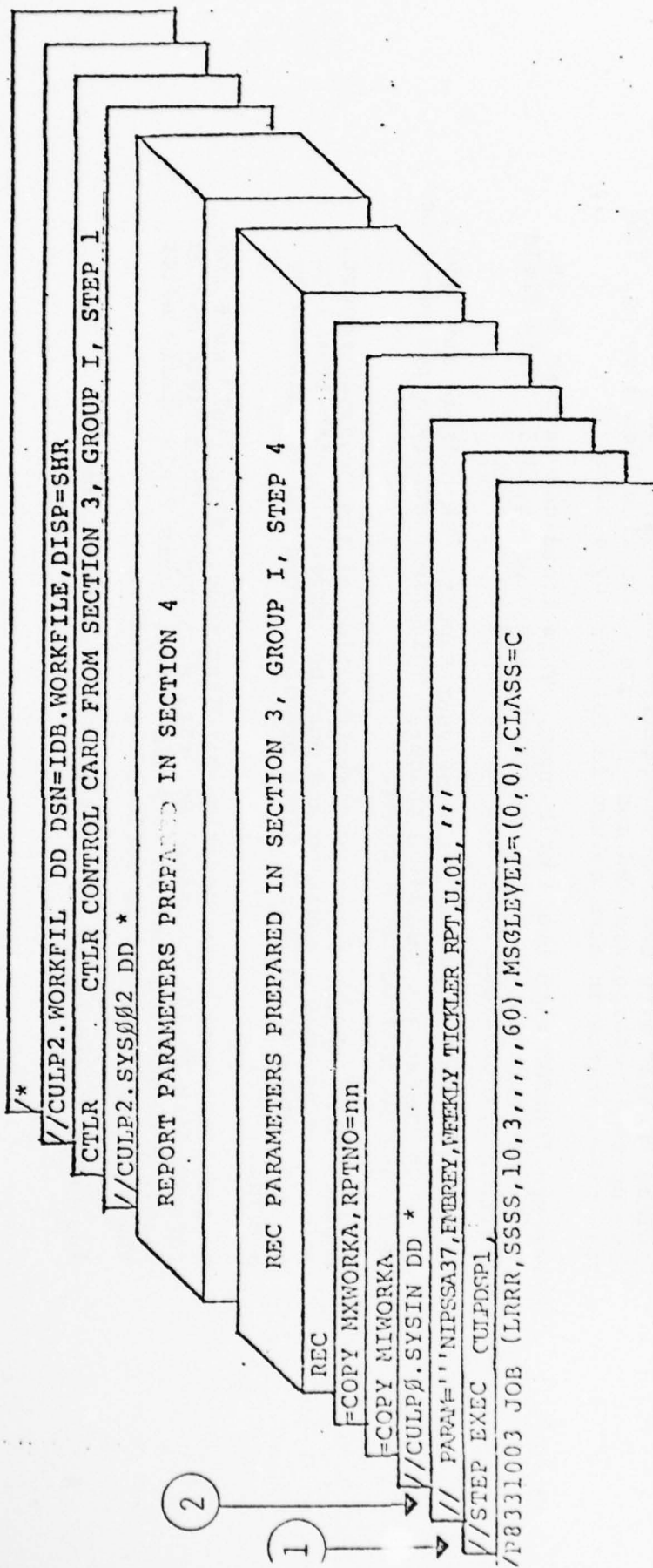
4 Replace "BBBBB" with the user's name. This field can have up to 16 characters and must be compatible with item 4 on pg. 3.5. This field starts after previous comma and is followed by a comma.

5 Replace "CCCCC" with the Title of your Report. This field may have up to 20 characters including spaces. This field starts after previous comma and is followed by a comma.

6 Replace "D" with the Security Classification of the produced report. This field is only 1 position and must be compatible with item 1 on pg. 3.5. This field starts after previous comma and is followed by a comma.

7 Replace "EE" with a two position handling code. This field must conform to DIAM 65-19, "Standard Security Markings." This field may be blank; however, you must include the comma. This field starts after the previous comma and is followed by a comma.

8 The PARAM statement must end with three single quotes.



CHANGE COMPUTER CONTROL DECK

1. The //STEP EXEC card change consists of replacing "CULPRIT" with "CULPDSPl,".
2. The PARAM card follows the //STEP EXEC card. The PARAM card is described on pg. 6.11.

NIPSSA37, EMBREY, WEEKLY TICKLER RPT, U, 1,  
TOTAL REPORT WAS SUCCESSFULLY STORED 790202

FIGURE 6.1

NIPSSA37, EMBREY, LIBCAT REPORT, U, 1,  
OVERFLOW DETECTED - FIRST 10 PAGES STORED 790202

FIGURE 6.2



### 6.3. CULPRIT Retrieval Errors.

Always check the last page of the listing which comes back from your CULPRIT run. It will tell you if your CULPRIT job was successfully run or had errors.

The first line will either display the PARAM card from the control deck or give the error:

PARAM CARD NOT FOUND - CHECK CONTROL DECK

If the error message is displayed, the PARAM card was missing from your control deck. Read Section 6.2 Change 2, include a PARAM card in your control deck and resubmit CULPRIT job.

The second line indicates the date the report was submitted and whether the complete report was successfully loaded. Figure 6.1 on previous page illustrates this. Also, if your CULPRIT report created more than 10 pages of output, a message will print out to indicate that only the first 10 pages of your report was loaded. An example of this is shown in Figure 6.2 on the previous page.

If any other errors appear on this last page of your report, it should be noted and taken immediately to your organization's ADP coordinator.

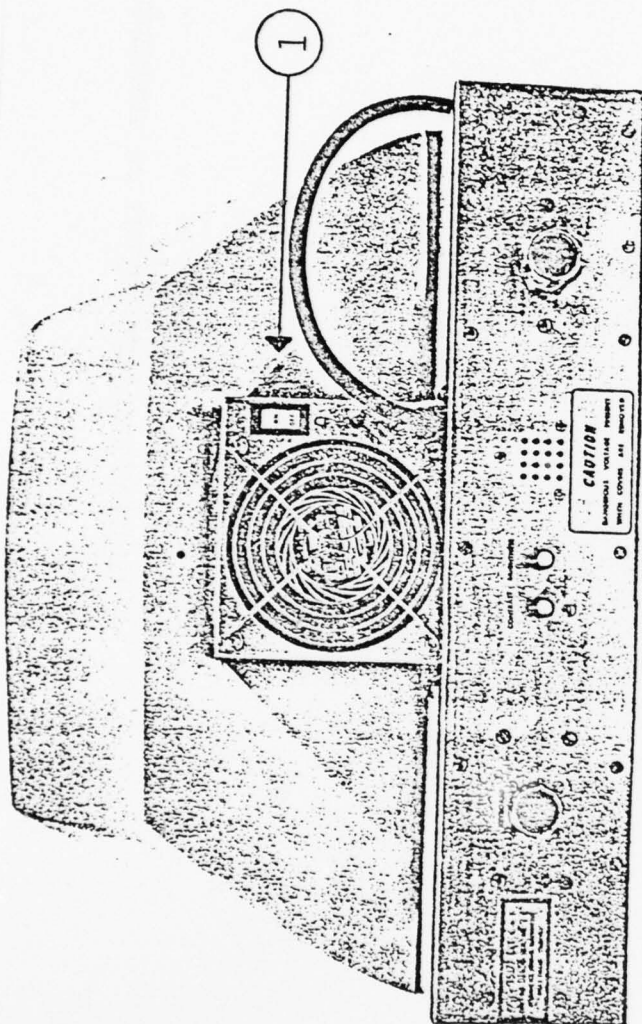
6.4. Part II - Display CULPRIT Output on Terminals.

This part will give a step by step procedure on how to operate the terminal and how to display your CULPRIT output on the terminal.

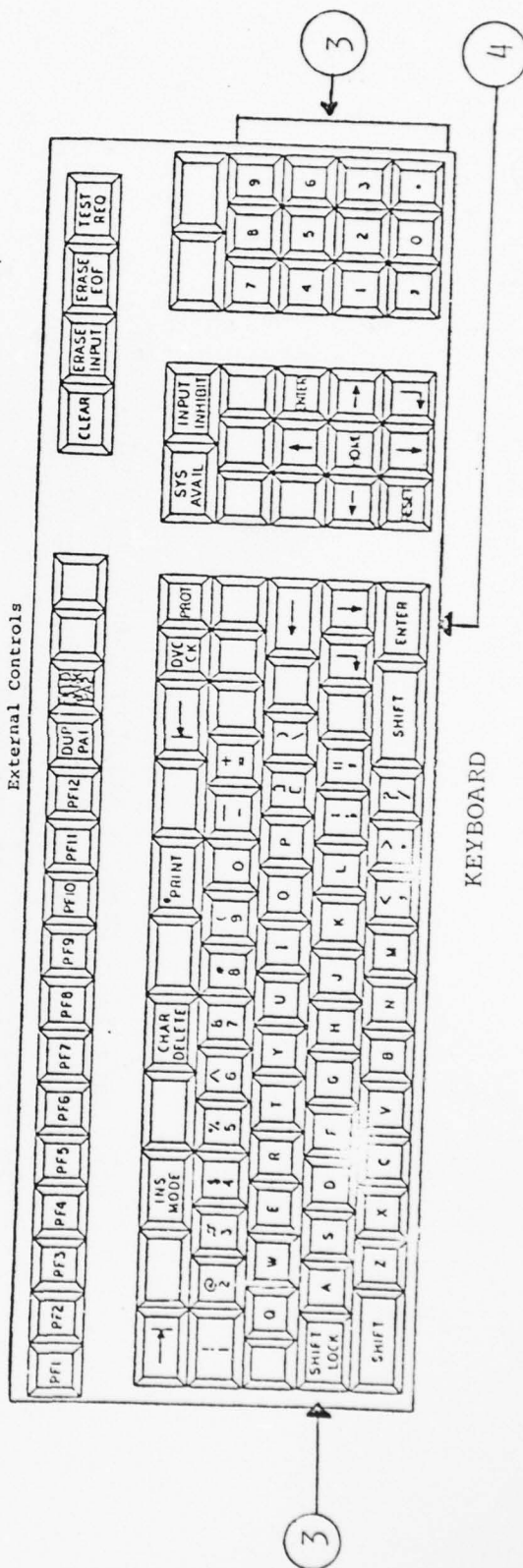
DISPLAY CULPRIT OUTPUT ON TERMINAL

PRECEDING PAGE BLANK

6.17(2.0)



External Controls



STEP 1.

CRT Terminal Log-on Procedures:

1. Turn on CRT Terminal by pressing ON the ON-OFF switch in back of the CRT Terminal screen.
2. Call (301) 763-2195 or AUTOVON 8-293-2195 and ask console operator to START LINE \_\_\_\_\_.
3. Press the "SHIFT LOCK" key. This will cause the Shift Lock key to light up and force the use of uppercase letters. Always use uppercase letters. Instead of shifting back to lowercase to key-in numbers, use the group of numbers on the far right of keyboard.
4. Press "ENTER" key.

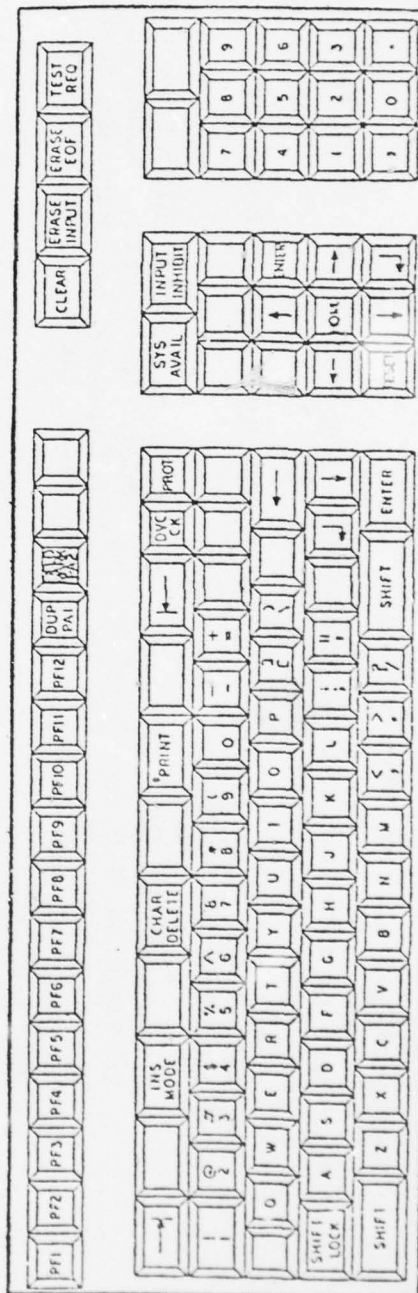
THE  
NAVAL INTELLIGENCE COMMAND  
ON-LINE SYSTEM  
(NICOLS)

IS READY TO SERVE YOU

PLEASE ENTER:

1 YOUR NAME -  
3 YOUR PASSWORD -

AND PRESS "ENTER" KEY





AD-A069 868

NAVAL INTELLIGENCE PROCESSING SYSTEMS SUPPORT ACTIVIT--ETC F/6 9/2  
INTEGRATED DATABASE DEVELOPMENT AND DESIGN GUIDE. VERSION 2.0.(U)  
MAY 79

UNCLASSIFIED

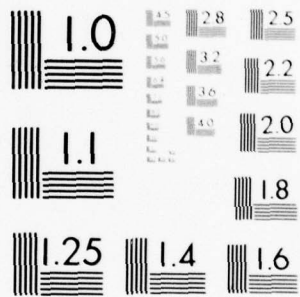
NIPSSA-UM-8000/2.0-5/79

NL

4 OF 7

AD  
A069868





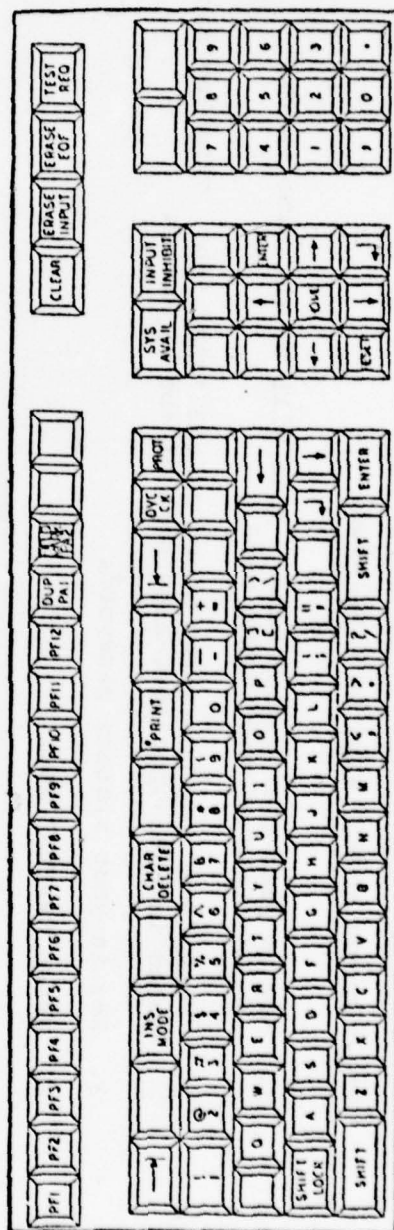
MICROCOPY RESOLUTION TEST CHART  
NATIONAL BUREAU OF STANDARDS-1963-A

STEP 2. The sign-on screen will be displayed.

1. Key in your user person name. This will be the same as the second field of your PARAM card.
2. Press "→" key hereafter known as the "TAB" key. This key can be found in the upper left-hand part of the keyboard.
3. Key in your systems password.
4. Press "ENTER" key.

## SHADOW II TRANSACTION CONTROL SYSTEM

ENTER YOUR NEXT TRANSACTION CODE DSP1



STEP 3. The transaction control screen will be displayed.

1. Key in "DSP1".
2. Press "ENTER" key.

6.23 (2.0)

DISPLAY CULPRIT OUTPUT ON TERMINAL



DSP1

WELCOME TO THE REPORT DISPLAY SUBSYSTEM OF NICOLS

PLEASE ENTER:

→ YOUR NAME - EMBREY

AND ORGANIZATION - NIPSSA37

→ TO OBTAIN AN INDEX OF YOUR REPORTS  
ENTER THE INDEX PAGE REQUIRED -  
AND PRESS THE PF4 KEY

→ TO OBTAIN A SPECIFIC REPORT  
ENTER THE REPORT NAME -  
THE REPORT DATE (YYMMDD) -  
THE SEQUENCE NUMBER -  
AND PRESS THE ENTER KEY

THE REPORT DISPLAY SUBSYSTEM MAY BE TERMINATED AT ANY TIME  
BY PRESSING THE PF2 KEY

PF1	PF2	PF3	PF4	PF5	PF6	PF7	PF8	PF9	PF10	PF11	PF12	DUP	PAI	PF13	PF14	PF15	PF16	PF17	PF18	PF19	PF20	PF21	PF22	PF23	PF24	PF25	PF26	PF27	PF28	PF29	PF30	PF31	PF32	PF33	PF34	PF35	PF36	PF37	PF38	PF39	PF40	PF41	PF42	PF43	PF44	PF45	PF46	PF47	PF48	PF49	PF50	PF51	PF52	PF53	PF54	PF55	PF56	PF57	PF58	PF59	PF60	PF61	PF62	PF63	PF64	PF65	PF66	PF67	PF68	PF69	PF70	PF71	PF72	PF73	PF74	PF75	PF76	PF77	PF78	PF79	PF80	PF81	PF82	PF83	PF84	PF85	PF86	PF87	PF88	PF89	PF90	PF91	PF92	PF93	PF94	PF95	PF96	PF97	PF98	PF99	PF100
										CLEAR ERASE INPUT ERASE EOP TEST REQ																																																																																											
SYS AVAIL INPUT INHIBIT SHIFTS → ← → ←										7 8 9 4 5 6 1 2 3 0																																																																																											
INS MODE CHAR DELETE PRINT DYS C.E.										0 1 2 3 4 5 6 7 8 9 A B C D E F G H I J K L M N O P Q R S T U V W X Y Z [ \ ]										SHIFT LOCK SHIFT ENTER																																																																																	

2

1

4

3

STEP 4.

The "WELCOME TO THE REPORT DISPLAY SUBSYSTEM OF NICOLS" screen will be displayed.

1. Key in user's name - must be identical to second field on PARAM card used to generate the report.
2. Press "TAB" key (⇠) on keyboard.
3. Key in user's organization acronym - must be identical to first field of PARAM card.
4. Now you have two choices: a) to obtain an index of your reports; or b) to obtain a specific report.

WELCOME TO THE REPORT DISPLAY SUBSYSTEM OF NICOLS

YOUR NAME - EMERY

AND ORGANIZATION - NIPSSA37

TO OBTAIN AN INDEX OF YOUR REPORTS

ENTER THE INDEX PAGE REQUIRED - 1

AND PRESS THE PF4 KEY

TO OBTAIN A SPECIFIC REPORT

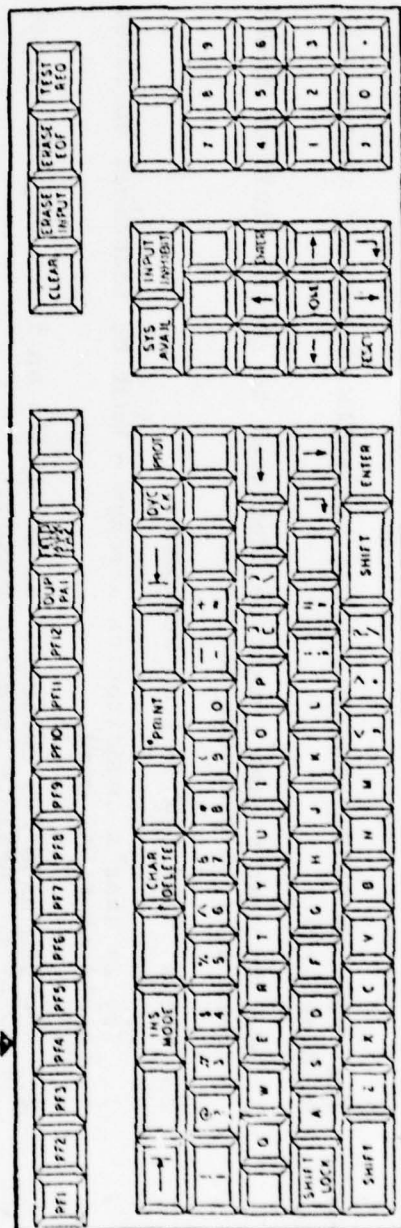
ENTER THE REPORT NAME - WEEKLY TICKLER RPT

THE REPORT DATE (YYMMDD) - 790206

THE SEQUENCE NUMBER - 001

AND PRESS THE ENTER KEY

THE REPORT DISPLAY SUBSYSTEM MAY BE TERMINATED AT ANY TIME BY PRESSING THE PF2 KEY



STEP 5.A

To obtain an index of your reports:

- A1. Press the "TAB" key.
- A2. Enter in "1" for the first page of your index.
- A3. Press PF4 key.

STEP 5.B

To obtain a specific report:

- B1. Press the "TAB" key twice.
- B2. Enter the report name - must be the same as third field of PARAM card.
- B3. Press "TAB" key once. (See A1.)
- B4. Enter the report date - This is the date your CULPRIT report was created. You must always enter the report date in (YYMMDD) format.
- B5. Enter the sequence number. When a report has been run only once on a particular day, the sequence number will be "1". The number increments when the report is run again on the same day without changing the PARAM card.
- B6. Press "ENTER" key.



DSP1

# INDEX OF AVAILABLE REPORTS

FOR USER NIPSSA37

EMBEY

PAGE 001 OF 001

ENTRY# -----REPORT-NAME----- RPT-SEQ RPT-DATE CLASS HAND #/PAGES COMPLETE?

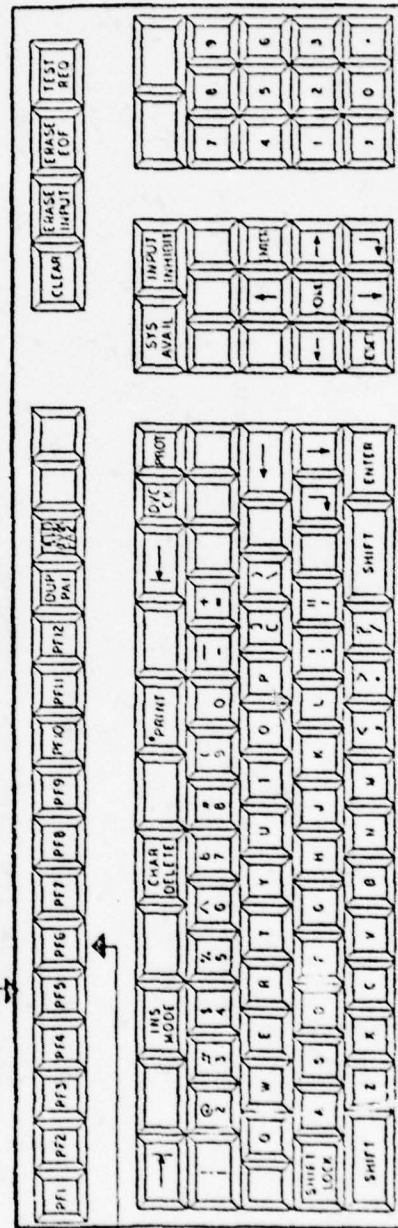
01	WEEKLY TICKLER RPT	001	02/06/79	U	1	003	YES
02	WEEKLY TICKLER RPT	001	02/08/79	U	1	003	YES
03	WEEKLY TICKLER RPT	001	02/13/79	U	1	006	YES
04	WEEKLY TICKLER RPT	002	02/13/79	U	1	006	YES
05	NULL REPORT	001	02/14/79	U	1	001	YES

PRESS THE PF5 KEY FOR NEXT INDEX PAGE  
 PRESS THE PF6 KEY FOR PREVIOUS INDEX PAGE  
 FOR A SPECIFIC REPORT, ENTER ENTRY# AND PRESS THE ENTER KEY

3A

1

2



3B

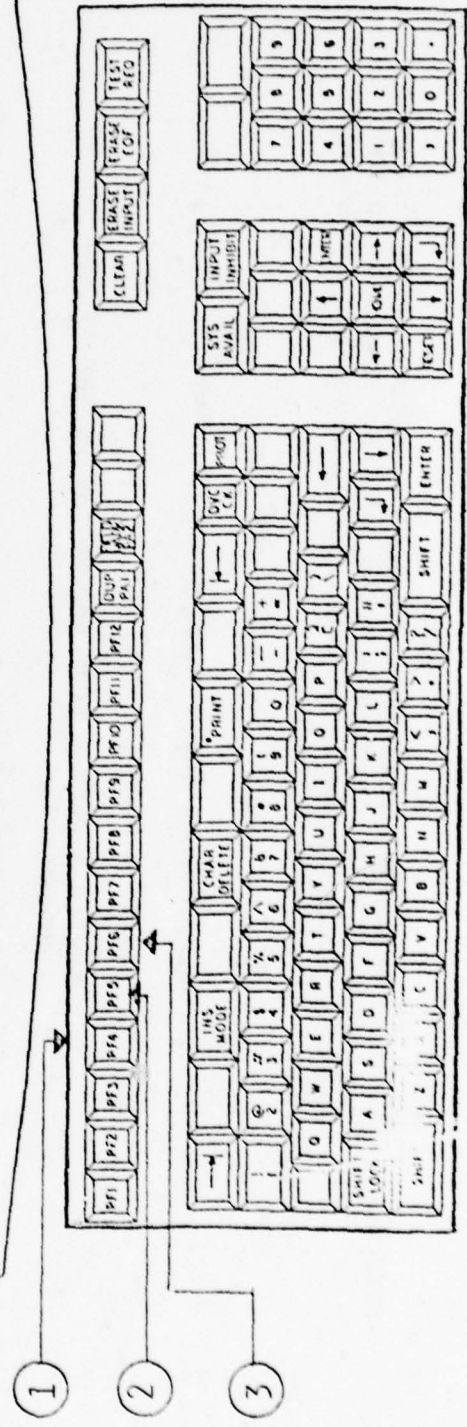


STEP 6.

If the PF4 key was pressed, an index of reports which have been generated by this user/organization will be displayed on the screen. There can be up to ten report entries per index page with multiple index pages.

1. To get next index page, press PF5 key.
2. To get previous index page, press PF6 key.
3. To display a specific report:
  - a. Key in a number from the ENTRY# column of the current index display.
  - b. Press "ENTER" key.

PERSON	DUE-DATE	SUB-UNIT	MESSAGE
EMBEY	78/12/08	514	DOCUMENT PROGRAM DSP1
EMBEY	78/12/31	514	WORK WITH BILL ON INPUT PROCESSES
FILIGENZI	78/11/03	512	REPORT TO NISC 360 HROWRE/SOFTWARE STAT EW
FILIGENZI	79/01/03	512	REPORT TO NISC 360 HROWRE/SOFTWARE STAT EW
FILIGENZI	79/01/03	35	NISC-ADP SUPPORT REQUEST
FILIGENZI	79/01/15	512	EST OF SECURITY CHG IMPACTS DUE
FILIGENZI	79/01/15	51	NISC-05 PROJ 21-02-03 MER. SHIP CHAR (MSC)
FILIGENZI	79/01/15	35	MONTHLY ACTIVITY SUMMARY DUE
FILIGENZI	79/01/15	35	NISC-ADP REQUEST FOR
FILIGENZI	79/02/28	512	MANAGEMENT OF TIME
FILIGENZI	79/03/03	512	REPORT TO NISC 360 HROWRE/SOFTWARE STAT EW
FILIGENZI	79/05/03	512	REPORT TO NISC 360 HROWRE/SOFTWARE STAT EW
POTTER	79/01/08	514	IDMS USER ASSN TAC MTG-BOSTON
POTTER	79/01/11	514	COORD MTG W/04 SUITLAND 0900



STEP 7.

If the "ENTER" key was pressed, the first page of your report will be displayed.

1. To return to report index screen, press PF4 key.
2. To get next page of your report, press PF5 key.
3. To get previous page of your report, press PF6 key.

If, at any time you want to end this session of the Report Display Subsystem (DSpl), press PF2 key. This will return to you the transaction control screen to enter in your next transaction.

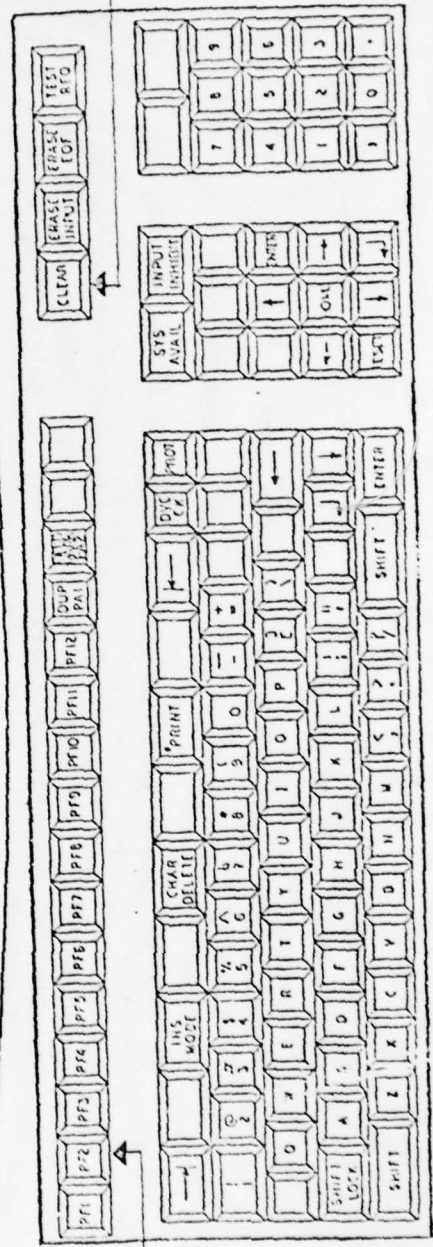
SHADOW II TRANSACTION CONTROL SYSTEM

ENTER YOUR NEXT TRANSACTION CODE

2A

2B

1



STEP 8.

CRT Terminal Sign-off Procedures:

1. Press PF2 key. This will display the transaction control screen.
2. Now you have two choices:
  - a. Enter next transaction as in STEP 3, or
  - b. Press "CLEAR" key to sign-off. The sign-on screen as in STEP 2 will be displayed. The terminal session is now complete.



6.5. Errors From Display Output.

This section will list the most common errors which may be displayed, their causes, and corrective action.

6.35 (2.0)

PRECEDING PAGE BLANK

ERRORS FROM DISPLAY OUTPUT

DSF1

# INDEX OF AVAILABLE REPORTS

FOR USER: NIPSSA37

EMBEY

PAGE 001 OF 001

ENTRY# -----REPORT-NAME----- RPT-SEQ RPT-DATE CLASS HAND #/PAGES COMPLETE?

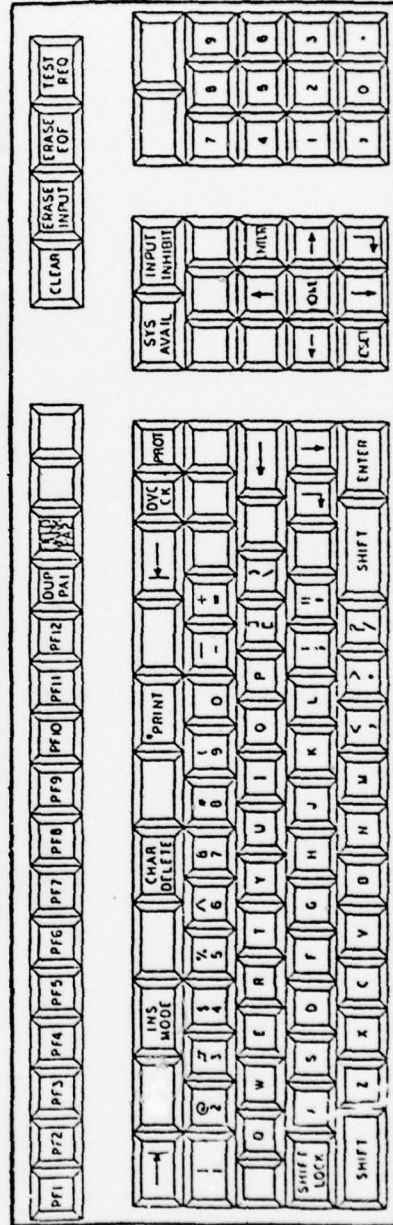
01	WEEKLY TICKLER RPT	001	02/06/79	U	1	003	YES
02	WEEKLY TICKLER RPT	001	02/08/79	U	1	003	YES
03	WEEKLY TICKLER RPT	001	02/13/79	U	1	006	YES
04	WEEKLY TICKLER RPT	002	02/13/79	U	1	006	YES
05	NULL REPORT	001	02/14/79	U	1	001	YES

PRESS THE PF5 KEY FOR NEXT INDEX PAGE

PRESS THE PF6 KEY FOR PREVIOUS INDEX PAGE

FOR A SPECIFIC REPORT, ENTER ENTRY# AND PRESS THE ENTER KEY

\* ERROR \* - 0000 - INVALID OPTION



ERROR	CAUSE	CORRECTIVE ACTION
AT FIRST PAGE OF INDEX	You were on the first page of index and pressed PF6 key.	Use other options at bottom of screen.
AT LAST PAGE OF INDEX	You were on last page of index and pressed PF5 key.	Use other options at bottom of screen.
INVALID OPTION	Did not press correct key or did not enter correct report number.	Re-read current step.
INVALID TRANSACTION	Questionable	Press "HOME" key, enter DSPl and press PF2 key. Go to Step 3 in 6.4.
NO NEXT OR PRIOR PAGE	Was on last page of report and pressed PF5 key or was on the first page of report and pressed PF6 key.	Use other options in the step.
NOTIFY NIPSSA DBA	There are possible problems with Database.	Print screen, if possible. Press PF2 key. Press "CLEAR" key. CONTACT NIPSSA DBA.
RECORD REQUESTED NOT FOUND	The information you keyed in was not correct.	Re-check fields for INVALID information. Re-read current step.

6.37(2.0)

ERRORS FROM DISPLAY OUTPUT

APPENDIX A

This appendix contains copies of all forms used by the working-file facility.

APPENDIX A

**SECURITY CLASS**

C	T	L	A
---	---	---	---

**HANDLING**

[illegible][illegible]

WORKING FILE IDENTIFIER	6	3
-------------------------	---	---

CTLR<sub>4</sub>

[illegible][illegible]



## MODIFICATION OF FILE CONTROL INFORMATION

A	Q	L	M
1	2	3	4

File Identifier							
S							12

Security Classification	Handling
14	16 17

[illegible]

	M	4
	M	3
	Q	2
A		1

[illegible][illegible]

# FILE DELETION FROM NICOLS

A	Q	L	I
1	2	3	4

File Identifier


5 12

CAUTION: This will erase your entire Working-File from the database.

GENERAL WORKING-FILE DATA INPUT FORM

[illegible]

**Note:** An asterisk (\*) in any position of the input form will cause the corresponding position in the working-file record to be blanked out. Any other character in any position of the input form will be stored in the corresponding position in the working-file record, replacing the character that previously existed in that position in the record.

[illegible]

GENERAL WORKING-FILE RECORD DEFINITION FORM


[illegible]

Record Identifier	17	21	25	29	32

THIS FORM IS FOR USE IN DEFINING  
CULPRIT QUERY POSITION DEFINITIONS  
IT IS NOT USED FOR DEFINING YOUR  
DATA ENTRY FORMAT.

[illegible]

# GENERAL WORKING-FILE - RECORD DELETION

<div> <div> <div>A</div> <div>Q</div> <div>M</div> <div>D</div> </div> <div>1 2 3 4</div> </div>	<div> <div>File Identifier</div> <div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> </div> <div>5 12</div> </div>	<div> <div>Record Identifier</div> <div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> </div> <div>13 28</div> </div>
<div> <div> <div>A</div> <div>Q</div> <div>M</div> <div>D</div> </div> <div>1 2 3 4</div> </div>	<div> <div>File Identifier</div> <div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> </div> <div>5 12</div> </div>	<div> <div>Record Identifier</div> <div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> </div> <div>13 28</div> </div>
<div> <div> <div>A</div> <div>Q</div> <div>M</div> <div>D</div> </div> <div>1 2 3 4</div> </div>	<div> <div>File Identifier</div> <div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> </div> <div>5 12</div> </div>	<div> <div>Record Identifier</div> <div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> </div> <div>13 28</div> </div>
<div> <div> <div>A</div> <div>Q</div> <div>M</div> <div>D</div> </div> <div>1 2 3 4</div> </div>	<div> <div>File Identifier</div> <div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> </div> <div>5 12</div> </div>	<div> <div>Record Identifier</div> <div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> </div> <div>13 28</div> </div>





# CULPRIT SORT AND TITLE STATEMENTS

Report  
Number

2	3
---	---

(Duplicate columns 2-3 for each line coded)

Lines/ Total/  
Page Detail

2	1
---	---

Report Sort Fields

16	20	25	30	35	40	45	50	55	60
----	----	----	----	----	----	----	----	----	----

2	8
---	---

Report Sort Fields continued

16	20	25	30	35	40	45	50	55	60
----	----	----	----	----	----	----	----	----	----


3	4
---	---

5	10	15	20	25	30	35	40	45	50	55
---	----	----	----	----	----	----	----	----	----	----

## CULPRIT REPORT DEFINITION FORM

Report Number 

2	3
---	---

Line	Type	
		

Line  
No.   5

Carr	10
Ctl	

(Duplicate columns 2-5 and 10 for each line coded)

[illegible]

# CULPRIT QUERY STATEMENTS

Report  
Number

2	3
---	---

7	4
---	---

(Duplicate columns 2-4 for each line coded)

Sequence  
Number

5	7
---	---

Process Instructions

8	10	15	20	25	30	35	40	45	50	55	58
---	----	----	----	----	----	----	----	----	----	----	----

5	7
---	---

8	10	15	20	25	30	35	40	45	50	55	58
---	----	----	----	----	----	----	----	----	----	----	----

5	7
---	---

8	10	15	20	25	30	35	40	45	50	55	58
---	----	----	----	----	----	----	----	----	----	----	----

5	7
---	---

8	10	15	20	25	30	35	40	45	50	55	58
---	----	----	----	----	----	----	----	----	----	----	----

5	7
---	---

8	10	15	20	25	30	35	40	45	50	55	58
---	----	----	----	----	----	----	----	----	----	----	----

5	7
---	---

8	10	15	20	25	30	35	40	45	50	55	58
---	----	----	----	----	----	----	----	----	----	----	----

5	7
---	---

8	10	15	20	25	30	35	40	45	50	55	58
---	----	----	----	----	----	----	----	----	----	----	----

5	7
---	---

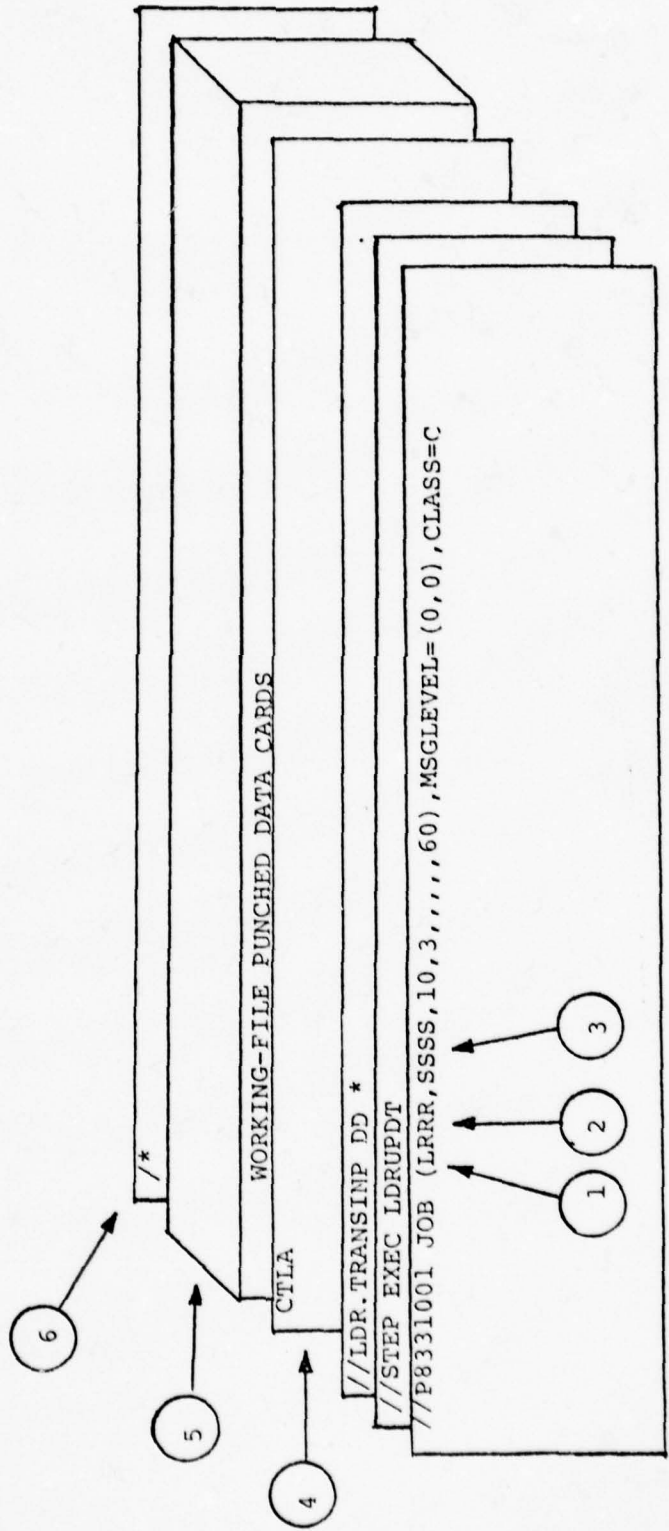
8	10	15	20	25	30	35	40	45	50	55	58
---	----	----	----	----	----	----	----	----	----	----	----

## APPENDIX B.1

This Appendix illustrates the computer control deck which is used to process updates to your working-file.

APPENDIX B.1





#### APPENDIX B.1 (WORKING-FILE MAINTENANCE)

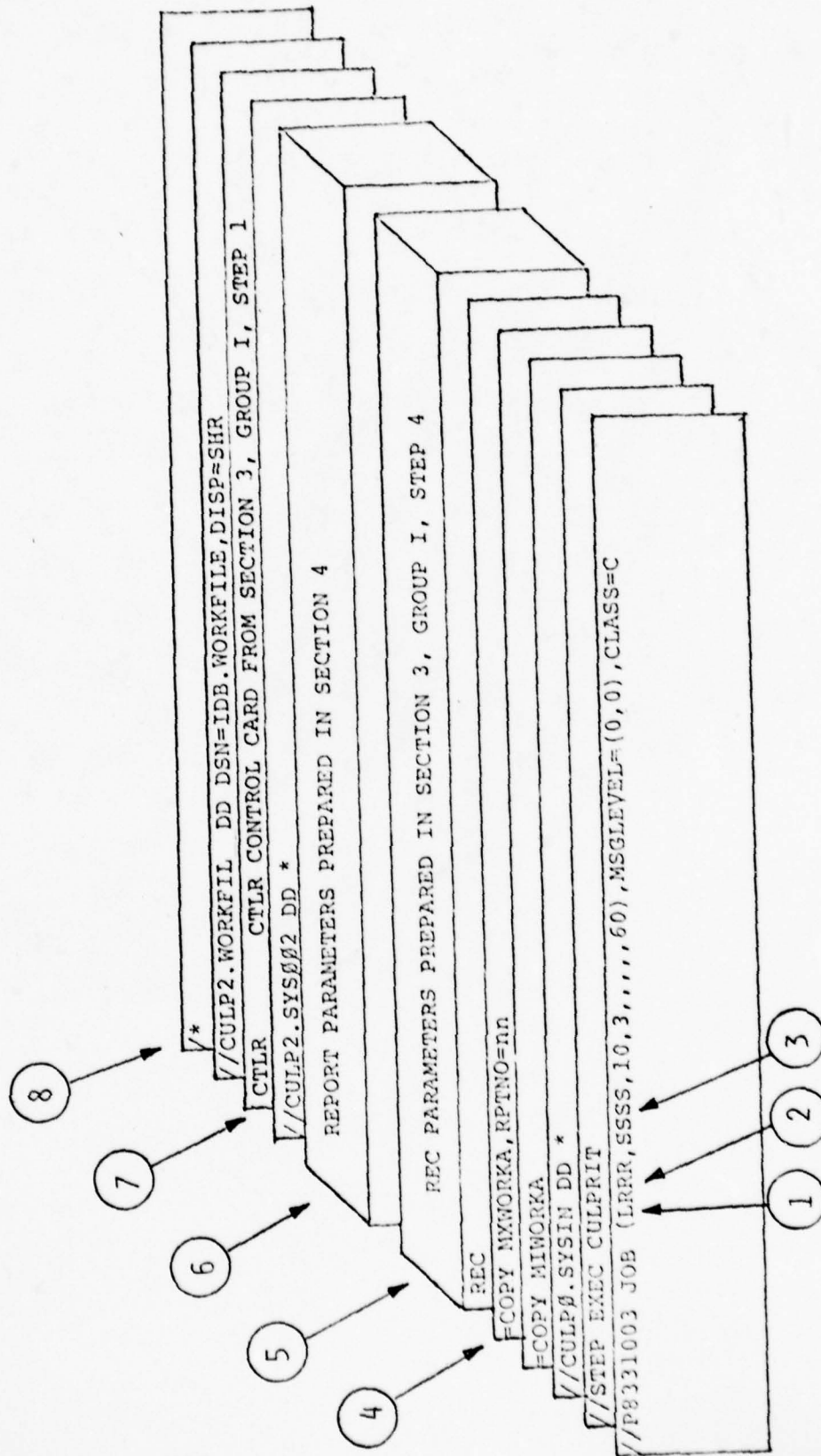
- 1     Replace the "L" with one of the following location codes:  
       H - Hoffman Building         S - FOB5 Suitland  
       M - Ft. Meade                F - Friendship Annex  
       P - Pentagon                  T - FOB2 Suitland
- 2     Replace "RRR" with a routing code assigned by the Database Administration staff or your organization ADP coordinator
- 3     Replace "SSSS" with the last four characters/digits of your office location identifier.
- 4     Insert the CTLA card which you prepared as part of identifying you to the NICOLS system.
- 5     Insert working-file punched card data entries. Entry cards may be placed in any order. The NICOLS system organizes the data into the proper order prior to updating the database.
- 6     The last card in the computer control deck must have "/"\* punched in the first two positions.

The "//" in the first three cards of the computer control deck begin in the first two positions of the card.

## APPENDIX B.2

This Appendix illustrates the computer control deck which is used to process reports against your working-file.

APPENDIX B.2



(1.1)

## APPENDIX B.2 (WORKING-FILE RETRIEVAL)

- 1      Replace the "L" with one of the following location codes:  

H- Hoffman Building	S- FOB5 Suitland
M- Ft. Meade	F- Friendship Annex
P- Pentagon	T- FOB2 Suitland
- 2      Replace "RRR" with a routing code assigned by the Database Administration staff or your organization ADP coordinator.
- 3      Replace "SSSS" with the last four characters/digits of your office location identifier.
- 4      Insert this card when you desire to print the security classification of the working-file on the top of your report. Change "nn" to the report number of the desired report.
- 5      Insert the REC parameters developed in Section 3, Group I, Step 4 here.
- 6      Insert the report parameters prepared in Section 4 here. Parameter decks for more than one report may be inserted. The decks should be placed so the first report to be produced is before the second, etc.  
NOTE: If security classification is required for multiple reports, step 4 must be repeated for each report where security printing is desired.
- 7      A single card with "=COPY MXWORKFL" punched beginning in position 1 will generate the file list report.
- 7      Insert a CTRLR control card which was prepared in Section 3, Group I, step 1.
- 8      The last card in the computer control deck must have a "/" punched in the first two positions.



USERS GUIDE  
FOR THE  
NAVAL INTELLIGENCE COMMAND  
MANAGEMENT INFORMATION SYSTEM  
(NIMIS)  
SUPPORTING  
THE NAVAL INTELLIGENCE COMMAND  
THROUGH THE  
NAVAL INTELLIGENCE COMMAND ON-LINE SYSTEM (NICOLS)

VERSION 3  
AUGUST 1978

#### FOREWORD

This Users Guide is designed to be used as a quick reference to assist you in using the NIMIS system. Each function is prepared separately with an illustration. The remainder of the Guide has been organized with the pages horizontal instead of vertical to make the instruction format more effective.

## TABLE OF CONTENTS

1.	INTRODUCTION.	
1.1.	Purpose.	e. Modify association of a person with a NIMIS project
2.	CAPABILITY DESCRIPTION	f. Record labor hours
2.1.	System Capabilities	g. Delete a person from the NIMIS database.
3.	CREATING AND MAINTAINING NIMIS INFORMATION	3.2.3. NIMIS Projects.
3.1.	Overview.	a. Create a NIMIS project.
3.2.	File Creation and Maintenance Functions	b. Modify short title/acronym
3.2.1.	NIMIS Organizations	c. Modify long title/short comments.
		d. Modify dates and status.
		e. Modify priority and classification.
		f. Modify resource hours remaining
		g. Add comments to NIMIS project
		h. Modify comments to NIMIS project
		i. Store references
		j. Modify references
		k. Delete references
		l. Create milestones
		m. Modify milestones
		n. Delete milestones
		o. Create sub-tasks
		p. Modify sub-tasks
		q. Delete sub-tasks
		r. Delete a NIMIS project
3.2.2.	NIMIS Personnel	3.2.4. NIC Tasks.
		a. Create a NIC task.
		b. Modify a NIC task
		c. Modify dates, status
		d. Modify subject, imis, point of contact

## TABLE OF CONTENTS

# TABLE OF CONTENTS (CONTINUED)

## 3.2.4. NIC Tasks.

- e. Modify short title
- f. Modify short comments
- g. Add comments to
- h. Modify comments to
- i. Modify priority, classification
- j. Associate with NIMIS project
- k. Separate association with a NIMIS project.
- n. All NIC Tasks summary
- o. Active NIC Tasks summary
- p. All NIC Tasks detail
- q. Active NIC Tasks detail
- r. Due date tickler of NIMIS projects
- s. Due date tickler of NIC Tasks
- 4.2. Special Queries Using Standard Report Formats.
- 4.3. Creating Special Report Formats.

## 4. QUERY AND REPORT PREPARATION

### 4.1. Standard Reports

- a. All NIMIS projects summary
- b. Active NIMIS projects summary
- c. Selected NIMIS project summary
- d. All NIMIS projects detail
- e. Active NIMIS projects detail
- f. Selected NIMIS projec detail
- g. All NIMIS projects supporting a requesting organization
- h. All NIMIS projects within organization hierarchy
- i. Active NIMIS projects within organization hierarchy
- j. NIMIS personnel supporting active projects
- k. NIMIS personnel resource distribution summary
- l. NIMIS personnel weekly summary
- m. NIMIS personnel assignment summary

## SECTION 1

### INTRODUCTION

#### 1.1. Purpose.

The Naval Intelligence Command Management Information System (NIMIS) is part of the Naval Intelligence Command On-line System (NICOLS). NIMIS has been established to provide NAVINTCOM management with a tool to assist in the direction and management of critical personnel and machine resources.

The NIMIS system has been designed to permit users to utilize the number of features which are necessary for their individual requirements. The system has broad project management capabilities.

Support for PERT- and Gantt-type structures is included for those who desire such a facility. The system allows association of multiple projects and organizations to duplicate actual conditions which occur during the processing of project tasking.

Finally, the system relates NIMIS projects to NIC tasks to improve the responsiveness of NAVINTCOM to its external users.

The Guide is designed for use by NAVINTCOM personnel at all levels who have a need for project management support.

## 1. INTRODUCTION



## SECTION 2

### CAPABILITY DESCRIPTION

#### 2.1. Facility Capabilities.

The NIMIS system has been designed

to:

1. Store associated organizations in the database.

2. Store personnel who work on NIMIS projects in the database.

3. Store NIMIS projects in the database, modify them, and associate them with other NIMIS projects.

4. Store NIC tasks in the database, modify them, and associate them with NIMIS projects.

5. Record labor hours against NIMIS projects and NIC tasks.

6. Record comments and status reports on NIMIS projects and NIC tasks.

7. Produce a wide variety of summary and detail reports on the progress of NIMIS projects and NIC tasks.

8. Provide a PERT- and Gantt support capability.

9. Permit the user to initiate special queries and reports against the database.

### 2. CAPABILITY DESCRIPTION

## SECTION 3

### CREATING AND MAINTAINING THE NIMIS DATABASE

#### 3.1. Overview.

This portion of the Users Guide describes the procedures for creating and maintaining the NIMIS database. Each function is described separately using an illustration of the standard input form and instructions for completing the form.

The instructions use some terms which may need clarification. The terms used throughout the Guide are:

Character - a letter, number, or special symbol, such as a "+", "-", or "\*".

Data Field - a collection or combination of characters identifying a specific item of information, such as date or social security number.

Field Position - a specific space or group of spaces defined on an input form.

Record - a collection of related data fields that have been defined on the standard input forms and which will be stored as a database record in the NICOLS system.

File - a collection of records belonging to a specific user.

Input Form - a form designed for the users of the NIMIS system which designates data field positions and which will be used for coding data.

#### 3.2. File Creation and Maintenance Functions.

The remainder of this section describes the various steps you should follow to create and maintain NIMIS records within the database. Illustrations of the standard NIMIS forms are provided with each update function. Copies of all forms for reproduction use are provided in Appendix A.

File creation and maintenance functions are separated into four groups:

1. NIMIS Organizations
2. NIMIS Personnel
3. NIMIS Projects
4. NIC Tasks

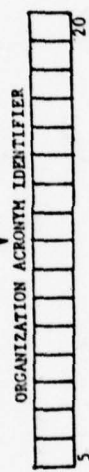
### 3. CREATING AND MAINTAINING THE NIMIS DATABASE

#### ENTERING AN ORGANIZATION INTO THE NIMIS DATABASE

This section defines the step required to establish an organization within the NIMIS database. This step is used primarily to add organizations which will become responsible for performance of NIMIS projects. Requesting organizations are automatically entered into the database if they were not previously entered.

ENTERING AN ORGANIZATION INTO THE NIMIS DATABASE

①



A	A	S	6
---	---	---	---

ENTERING AN ORGANIZATION INTO THE NIMIS DATABASE

AAPS INPUT LINE

Item	Field Position on		Data Field Name and Description
	Input Form		
1	5-20		ORGANIZATION ACRONYM IDENTIFIER. Enter the acronym of the organization to be entered into the NIMIS database. The beginning of the acronym should be placed in the left-most position of the field.

ENTERING AN ORGANIZATION INTO THE NIMIS DATABASE



CHANGE THE ACRONYM/IDENTIFIER OF A NIMIS ORGANIZATION

This section describes the steps required to change the acronym/identifier of an organization related to the NIMIS database.

CHANGE THE ACRONYM/IDENTIFIER OF A NIMIS ORGANIZATION

1  
CHANGE THE ACRONYM/IDENTIFIER OF A NIMIS ORGANIZATION

1  
A A Q I  
4

ORIGINAL ORGANIZATION NAME/ACRONYM  
3 20

NEW ORGANIZATION NAME/ACRONYM  
35 50

2

# ENTERING AN ORGANIZATION INTO THE NIMIS DATABASE

## AAPS INPUT LINE

Item	Field Position on Input Form	Data Field Name and Description
1	5-20	ORGANIZATION ACRONYM IDENTIFIER. Enter the acronym of the organization to be entered into the NIMIS database. The beginning of the acronym should be placed in the left-most position of the field.

ENTERING AN ORGANIZATION INTO THE NIMIS DATABASE

CHANGE THE ACRONYM/IDENTIFIER OF A NIMIS ORGANIZATION

This section describes the steps required to change the acronym/identifier of an organization related to the NIMIS database.

CHANGE THE ACRONYM/IDENTIFIER OF A NIMIS ORGANIZATION

CHANGE THE ACRONYM/IDENTIFIER OF A NMIS ORGANIZATION

1

1 A A Q I 4

ORIGINAL ORGANIZATION NAME/ACRONYM 5 20

NEW ORGANIZATION NAME/ACRONYM 35 50

2



# CHANGE THE ACRONYM/IDENTIFIER OF A NIMIS ORGANIZATION

## AAQI INPUT LINE

Item	Field Position on Input Form	Data Field Name and Description
1	5-20	ORIGINAL ORGANIZATION NAME/ACRONYM. Enter the name or acronym of the organization as it now appears in the NIMIS database.
2	35-50	NEW ORGANIZATION NAME/ACRONYM. Enter the name or acronym of the organization as it is to appear in the NIMIS database.

## CHANGE THE ACRONYM/IDENTIFIER OF A NIMIS ORGANIZATION

#### ASSOCIATE TWO ORGANIZATIONS IN THE NIMIS DATABASE

NIMIS has the ability to relate organizations. This is useful when the structure of participating organizations is to be used for reporting project activity. The relationship of a NIMIS organization to those subsidiary to it can be established in such a way that summary reports can be produced covering an organization and all of its subsidiaries. This permits, for example, commanding officers to review the resource utilization of subsidiary departments while permitting the departments to review subsidiary divisions, etc. NIMIS projects can thus be managed by the lowest organizational level tasked to perform the work while upper management can review overall resource utilization without further data input.

This section describes the steps required to relate two NIMIS organizations to one another.

ASSOCIATE TWO ORGANIZATIONS IN THE NIMIS DATABASE

1  
ASSOCIATE TWO ORGANIZATIONS IN THE NIMIS DATABASE

1  
A A 0 I  
4

5 20  
PRINCIPAL ORGANIZATION ACRONYM IDENTIFIER

35 50  
SUBSIDIARY ORGANIZATION ACRONYM IDENTIFIER

2

# ASSOCIATE TWO ORGANIZATIONS IN THE NIMIS DATABASE

## AAQI INPUT LINE

Item	Field Position on Input Form	Data Field Name and Description
1	5-20	PRINCIPAL ORGANIZATION ACRONYM IDENTIFIER. Enter the acronym of the principal organization in the relationship to be established. The organization acronym should begin in the left-most position of the field.
2		SUBSIDIARY ORGANIZATION ACRONYM IDENTIFIER. Enter the acronym of the subsidiary organization in the relationship to be established. The organization acronym should begin in the left-most position of the field.

ASSOCIATE TWO ORGANIZATIONS IN THE NIMIS DATABASE

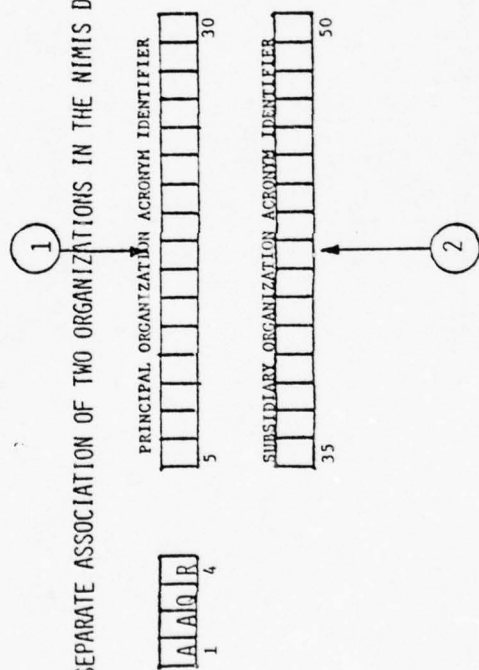
SEPARATE ASSOCIATION OF TWO ORGANIZATIONS IN THE NIMIS DATABASE

This section describes the steps required to separate the association of two organizations within the NIMIS database.

SEPARATE ASSOCIATION OF TWO ORGANIZATIONS IN THE NIMIS DATABASE



### SEPARATE ASSOCIATION OF TWO ORGANIZATIONS IN THE NIMIS DATABASE



# SEPARATE ASSOCIATION OF TWO ORGANIZATIONS IN THE NIMIS DATABASE

## AAQR INPUT LINE

Item	Field Position on Input Form	Field Data Name and Description
1	5-30	PRINCIPAL ORGANIZATION ACRONYM IDENTIFIER. Enter the acronym of the principal organization which will be separated from a subsidiary. Enter the acronym in the left-most positions of the field.
2	35-50	SUBSIDIARY ORGANIZATION ACRONYM IDENTIFIER. Enter the acronym of the subsidiary organization which is to be separated from its principal. Enter the acronym in the left-most positions of the field.

SEPARATE ASSOCIATION OF TWO ORGANIZATIONS IN THE NIMIS DATABASE

REMOVING AN ORGANIZATION FROM THE NIMIS DATABASE

On occasion, it is necessary to remove an organization from the NIMIS database. This section describes the removal function.

CAUTION: This step physically removes the record containing the organization from the database and makes it inaccessible for future processing.

REMOVING AN ORGANIZATION FROM THE NIMIS DATABASE

①

A	A	P	D
---	---	---	---

[illegible]

[CAUTION: THIS OPTION WILL PHYSICALLY DELETE THE ORGANIZATION RECORD FROM THE NIMIS DATABASE. ]

REMOVING AN ORGANIZATION FROM THE NIMIS DATABASE

AAPD INPUT LINE

Item	Field Position on Input Form	Data Field Name and Description
1	5-20	ORGANIZATION ACRONYM IDENTIFIER. Enter the acronym name of the organization to be deleted from the NIMIS database. The name of the organization begins in the left-most position of the field.

REMOVING AN ORGANIZATION FROM THE NIMIS DATABASE

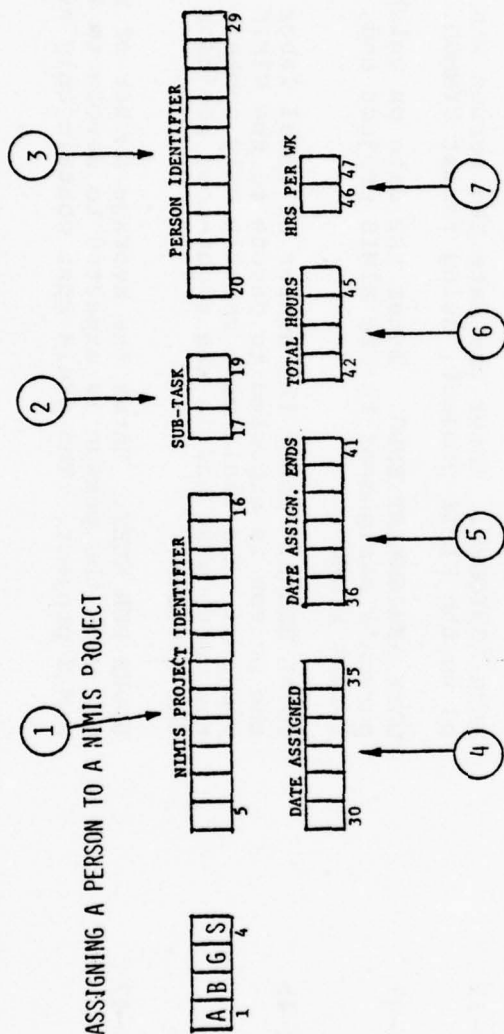


#### ASSIGNING A PERSON TO A NIMIS PROJECT

This section provides the steps necessary to assign a person to a NIMIS project. The user is provided with the capability to define the time frame in which a person is assigned to the project, the total number of hours, and the average hours per week.

#### ASSIGNING A PERSON TO A NIMIS PROJECT

# ASSIGNING A PERSON TO A NIMIS PROJECT



# ASSIGNING A PERSON TO A NIMIS PROJECT

## ABGS INPUT LINE

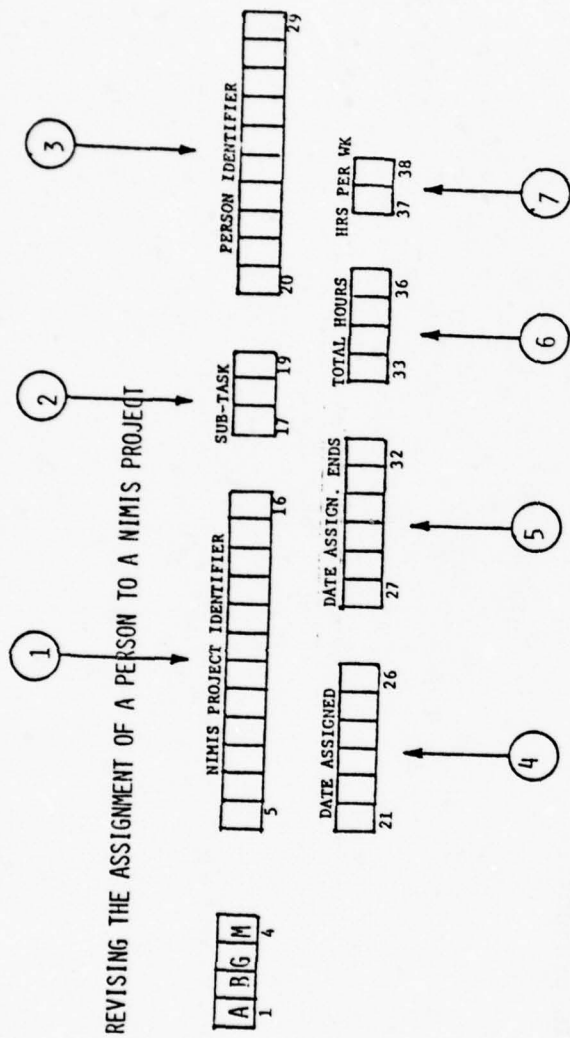
Item	Field Position on Input Form	Data Field Name and Description
1	5-16	NIMIS PROJECT IDENTIFIER. Enter the identifier of the NIMIS project to which a person is to be assigned.
2	17-19	SUB-TASK. Enter the three digit sub-task identifier, if applicable. If no sub-task is assigned, leave the field blank.
3	20-29	PERSON IDENTIFIER. Enter the identifier of the person to be assigned. If the identifier is the person's social security number, enter the number in the right-most nine positions of the field.
4	30-35	DATE ASSIGNED. Enter the date the person was assigned to the NIMIS project, using format YYMMDD.
5	36-41	DATE ASSIGNMENT ENDS. Enter the date on which the person's assignment to the NIMIS project end, using format YYMMDD.
6	42-45	TOTAL HOURS. Enter the number of total labor hours the person is expected to devote to the NIMIS project. The field must contain only numbers and the value must end in the right-most position of the field.
7	46-47	HOURS PER WEEK. Enter the average number of hours per week the person is expected to devote to the NIMIS project. The field must contain only numbers.

## ASSIGNING A PERSON TO A NIMIS PROJECT

REVISING THE ASSIGNMENT OF A PERSON TO A NIMIS PROJECT

This section provides the capability to change the conditions of assignment of a person to a NIMIS project.

REVISING THE ASSIGNMENT OF A PERSON TO A NIMIS PROJECT





# REVISING THE ASSIGNMENT OF A PERSON TO A NIMIS PROJECT

## ABGM INPUT LINE

Item	Field Position on Input Form	Data Field Name and Description
1	5-16	NIMIS PROJECT IDENTIFIER. Enter the identifier of of the NIMIS project to which the person is assigned.
2	17-19	SUB-TASK. Enter the sub-task identifier, if applic- able. Leave the field blank if no sub-task is used.
3	20-29	PERSON IDENTIFIER. Enter the identifier of the person assigned to the NIMIS project. If the ident- ifier is the person's social security number, enter the number in the right-most nine positions of the field.
4	30-35	DATE ASSIGNED. Enter the revised assignment date using the format YYMMDD.
5	36-41	DATE ASSIGNMENT ENDS. Enter the revised date on which the person's assignment ends, using format YYMMDD.
6	42-45	TOTAL HOURS. Enter the number of total labor hours that the person is expected to devote to the NIMIS project. The field must contain only numbers and the value must end in the right-most position of the field.
7	46-47	HOURS PER WEEK. Enter the revised number of average hours per week the person is expected to devote to the NIMIS project. The field must contain only numbers.

NOTE: Fields 1, 2, and 3 are required. Fields 4, 5, and 6 are completed only if that  
value is to be changed in the database.

REVISING THE ASSIGNMENT OF A PERSON TO A NIMIS PROJECT

PROJECT LABOR HOUR ENTRY

This section describes the form used to record labor hours charged by assigned personnel to various NIMIS projects. The form is completed by personnel on a pre-defined period, usually weekly.

PROJECT LABOR HOUR ENTRY

## MINIMIS PROJECT LABOR HOUR ENTRY

This information is duplicated into all data cards on this page.]

[illegible]

## PROJECT LABOR HOUR ENTRY

## ABPS INPUT LINE

Item	Field Position on		Data Field Name and Description
	Input	Line	
1	4-14		PERSON IDENTIFIER. Enter the individual identifier of the person charging hours to a NIMIS project. If the social security number of the person is used, it is placed in the right-most nine positions of the field.
2	30-35		PERIOD DATE. Enter the date when the reporting period began, using the format YYMMDD.
3	15-26		NIMIS PROJECT IDENTIFIER. Enter the identifier of the NIMIS project being charged.
4	27-29		SUB-TASK. Enter the sub-task number of the sub-task within the NIMIS project which is being charged. If no sub-task is assigned, leave the field blank.
5	36-39		REGULAR HOURS. Enter the number of whole hours charged to the NIMIS project. The field must contain numbers only and the value is entered into the right-most positions of the field.
6	40-43		OVERTIME HOURS. Enter the number of overtime hours charged to the NIMIS project. The field must contain only numbers and the value must be entered into the right-most positions of the field.
7	44-80		REMARKS. Enter pertinent remarks about effort expended on the NIMIS project. Remarks are free-form and are entered into the NIMIS database as level 8 remarks records.

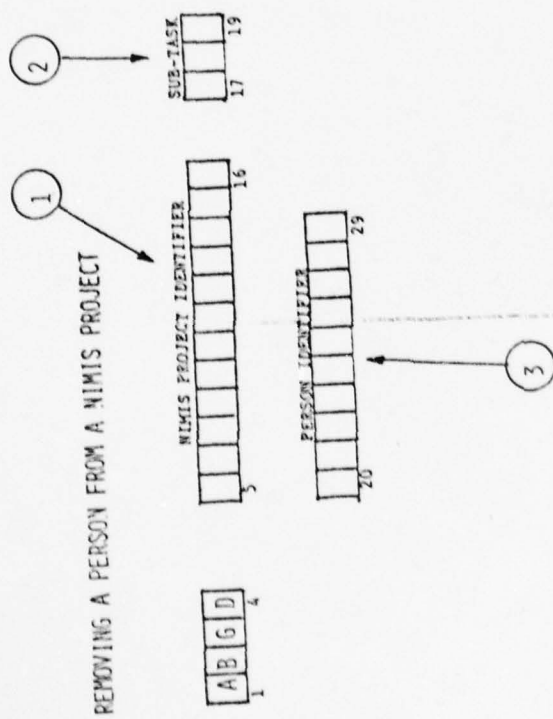
PROJECT LABOR HOUR ENTRY

REMOVING A PERSON FROM A NIMIS PROJECT

This section describes how to alter a person's assignment by removing them from a NIMIS project.

REMOVING A PERSON FROM A NIMIS PROJECT





# REMOVING A PERSON FROM A NIMIS PROJECT

## ABGD INPUT LINE

Item	Field Position on Input Form	Data Field Name and Description
1	5-16	NIMIS PROJECT IDENTIFIER. Enter the identifier of the NIMIS project from which the person is being reassigned.
2	17-19	SUB-TASK. Enter the sub-task identifier. Leave the field blank if not applicable.
3	20-29	PERSON IDENTIFIER. Enter the identifier of the person being reassigned. If the social security number is used as a person identifier, enter it in the right-most nine positions of the field.

REMOVING A PERSON FROM A NIMIS PROJECT

## APPENDIX G

### INTEGRATED DATA DICTIONARY (IDD) SYNTAX

This appendix contains excerpts from the users guide for the Integrated Data Dictionary (IDD) product developed and marketed by Cullinane Corporation. The sections presented in this appendix relate only to the syntax of IDD entry statements.

The material in this appendix is copyrighted by Cullinane Corporation.

## SECTION 3

### DDDL COMPILER AND LANGUAGE SPECIFICATIONS

#### DDDL COMPILER PROGRAM OPERATION

The Data Dictionary Definition Language (DDDL) is the source language used to supply the Dictionary/Directory with definitions of and relationships between dictionary entities.

The DDDL Compiler accepts Data Dictionary Definition Language input and uses IDMS to update the Dictionary/Directory Network.

The Dictionary/Directory Network is an IDMS database. At installation time, IDMSDIRL loads the network structure to a user-specified disk extent. Then, the DDDL compiler updates the Dictionary/Directory with the default level numbers, certain classes and attributes, and the DREPORT modules.

The Data Dictionary Activity List shows DDDL source submitted, along with any error messages generated for the run.

Figure 3-1 illustrates the components described above.

#### DATA DICTIONARY ACTIVITY LIST

The Data Dictionary Activity List shows DDDL source submitted, along with any error messages generated for the run. (A list of error messages may be found in Appendix A.) The Data Dictionary Activity List may be formatted to skip a specified number of lines between DDDL source statements or to advance to the top of the next page before printing the next statement by the insertion of SKIP and EJECT cards at appropriate intervals in the DDDL input source deck:

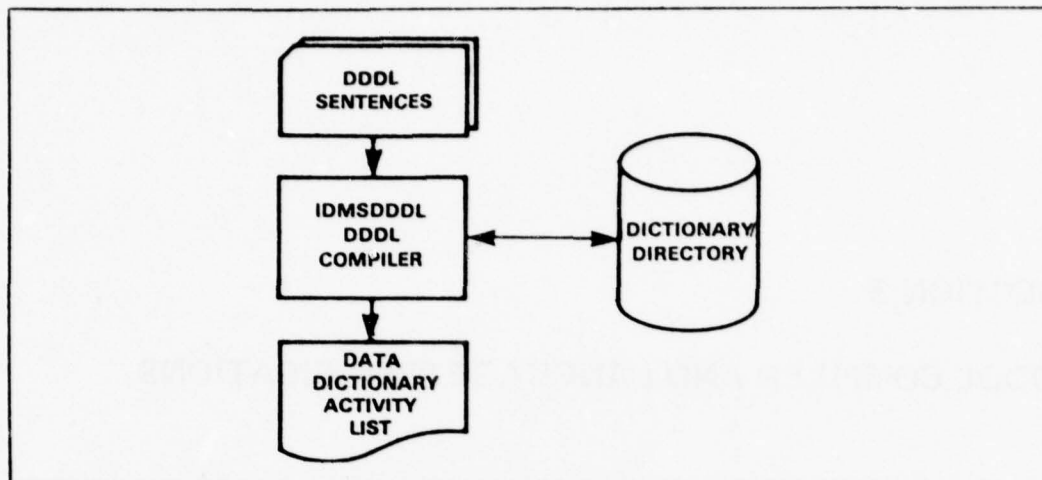


FIGURE 3-1. MAINTENANCE OF THE DICTIONARY PORTION OF THE DICTIONARY/DIRECTORY NETWORK

- *SKIPinteger*

*Integer* may be 1, 2, or 3, and specifies the number of lines to be skipped. The command must be contained within columns 8-72.

- EJECT

This command causes the printer to advance to the top of the next page. The command must be contained within columns 8-72.

The SKIP and EJECT cards do not appear in the Data Dictionary Activity List, a sample of which is presented in Figure 3-2.

## DDDL LANGUAGE SPECIFICATIONS

The Data Dictionary Definition Language (DDDL) is the source language used to populate the Dictionary/Directory Network with definitions of and relations between dictionary entries.

The basic unit of input into the DDDL Compiler is the sentence dealing with an entity occurrence in the IDD. Each sentence begins with a verb (ADD, MODIFY, DELETE) designating the function to be performed. This is followed by the NAME clause, which identifies the entity category (SYSTEM, SUBSYSTEM, USER, FILE, PROGRAM, MODULE, ELEMENT, RECORD, CLASS, or ATTRIBUTE) and specifies either a user-supplied primary name or the system default name for



10MB00DL 1.00	CULLENHANE CORPORATION INTEGRATED DATA DICTIONARY COMPILER DATA DICTIONARY ACTIVITY LIST	DATE 03/16/77	TIME 13143025	PAGE 0008
0052	ADD ELEMENT			
0053	NAME IS ZIP-CODE			
0054	ELEMENT DESCRIPTION IS CODE			
0055	PIC X(4)			
0056	COMMENTS 'US ZIPCODES ARE FOUND LEFT-JUSTIFIED IN THIS			
0057	'FIELD. 4TH CHARACTER ACCOMMODATES			
0058	'CANADIAN & BRITISH MAILING CODES'.			
0059	ADD ELEMENT NAME IS COUNTRY			
0060	PIC X(12)			
0061	COMMENTS 'A BLANK COUNTRY IS ASSUMED = USA'.			
0062	ADD ELEMENT			
0063	NAME IS ADDRESS			
0064	SUBORDINATE ELEMENTS ARE			
0065	STREET-ADDRESS			
0066	CITY			
0067	STATE-CODE			
0068	ZIP-CODE			
0069	COUNTRY.			
0070	ADD ELEMENT			
0071	NAME IS TEL-AREA-CODE			
0072	ELEMENT DESCRIPTION IS CODE			
0073	PIC XX.			
0074	ADD ELEMENT NAME IS TEL-NUMBER			
0075	PIC X(7)			
0076	COMMENTS '3 DIGIT EXCHANGE + 4 DIGIT NUMBER'.			
0077	ADD ELEMENT			
0078	NAME IS TEL-EXTENSION			
0079	PIC X(4)			
0080	COMMENTS 'LEFT JUSTIFIED IF 2 OR 3 CHARACTER'.			
0081	ADD ELEMENT			
0082	NAME IS TELEPHONE.			
0083	SUBORDINATE ELEMENTS ARE			
0084	TEL-AREA-CODE			
0085	TEL-NUMBER			
0086	TEL-EXTENSION.			
0087	ADD ELEMENT			
0088	NAME IS DATE-OF-ENTRY			
0089	ELEMENT DESCRIPTION IS DATE			
0090	PIC X(4)			
0091	COMMENTS 'DATE OF ENTRY (MMYY) ONTO DATABASE'.			
0092	ADD ELEMENT			
0093	NAME IS ORIGIN-CODE			
0094	ELEMENT DESCRIPTION IS CODE			
0095	PIC XX			
0096	COMMENTS 'CODES SUPPLIED BY CULLENHANE CORPORATION'.			
0097	ADD ELEMENT NAME IS SIC			

FIGURE 3-2. DATA DICTIONARY ACTIVITY LIST

the entity occurrence. Subsequent clauses are optional. Each consists of a *required* keyword/keyword expression (SECURITY, COMMENTS, ELEMENT DESIGNATOR, etc.), an *optional* keyword that may be included to clarify syntax and thus enhance readability (IS, ARE, BY, etc.), and a corresponding user-supplied or system default expression (name, description, or other alphanumeric or numeric literal). A period followed by a space terminates the sentence.

ADD ELEMENT  
NAME IS ACTION-CODE  
PIC X.

MODIFY ELEMENT  
NAME IS ACTION-CODE  
COMMENTS 'INTRODUCED 10/12/78'.

DELTE CLASS  
NAME IS ACTION-CODE.

The syntax notation for MODIFIES of multiple entry clauses indicates that the INCLUDE and EXCLUDE words are optional. If neither word is used, INCLUDE is assumed. The rules for the INCLUDE function are the same as those for the ADD function explained for the entity.

Treatment of the DDDL at the detail level may be arranged conveniently under three headings:

- Conventions
- PREPARED/REVISED BY Clause
- DDDL Sentence Presentation Sequence
- DDDL Sentence Syntax Displays

## CONVENTIONS

The following discussion covers the acceptable coding format, syntax notation, character set, and word types:

### Coding Format

The DDDL input record format is an 80 character card-image. The following table describes the input record format.

TABLE 3-1. INPUT RECORD FORMAT

Columns (s)	Contents
1 - 6	Line sequence number - optional; ignored by DDDL Compiler
7	Asterisk - indicates entry is documentary only; line is passed to the output listing without being compiled
	Hyphen - used as a continuation character for the COMMENTS and ELEMENT DEFINITION clauses only.
8 - 72	Reserved for user DDDL entries, and SKIP and EJECT commands.
73 - 80	User text area - ignored by the DDDL compiler; entries may contain any user documentation.

All clauses except the COMMENTS and ELEMENT DEFINITION clauses may be continued from one line to the next without a continuation character. However, words may *not* be split between lines. The following are two examples of acceptable DDDL input for the same sentence.

Example 1 (card 1 of 1) ADD RECORD NAME IS CUSTOMER.

Example 2 (card 1 of 2) ADD RECORD  
(card 2 of 2) NAME IS CUSTOMER.

The COMMENTS clause and the ELEMENT DEFINITION clause may consist of any number of cards. Comments may be associated with any Dictionary entity. The element definition is applicable only to elements. Rules for coding both clauses are the same.

The user-supplied literal must be enclosed in quotes if embedded blanks or special characters are included. Card columns 8-72 only may be used to contain the literal, including bounding quotes. The continuation character (hyphen in column 7) must be used on second and subsequent cards when more than one card is entered. If the ending quote is omitted, the contents of that card up to column 72 will be included as part of the literal. To include a quote within a comment or definition, enter two quote characters in succession.

Examples of clauses which may consist of multiple cards follow.

```
ADD ELEMENT
NAME IS ACTION=CODE
ELEMENT DESCRIPTION IS CODE
PIC X
COMMENTS ' + IS AN ADD, - IS A DELETE, = IS A MODIFY',
```

The following example shows how to add comments with multiple cards and how to code the quote when it is included in the comments.

```
ADD ELEMENT
NAME IS COMPTROL=KEY
ELEMENT DESIGNATOR IS CALCKEY
PIC X(4)
COMMENTS ' CALC KEY FOR OOK COMPTROL RECORD
          'VALUE='ICHP1''',
ADD ELEMENT
NAME IS NEXT=NUMBER
```

To modify comments already stored in the Dictionary, include the COMMENTS clause with the appropriate MODIFY command. Previously specified comments, if any, are replaced by those entered at this time. All comments must be resubmitted - individual comments lines *cannot* be modified. The NULL operation deletes all comments for an entity.

The following example shows how to replace comments for an entity.

```

MODIFY ELEMENT
NAME IS ACTION-CODE
COMMENTS '+ IS AN ADD; - IS A DELETE; * IS A MODIFY!
-      'INVALID ACTION CODE PRODUCES UNPREDICTABLE'
-      'RESULTS',

```

The following example shows how to delete comments for an entity.

```

MODIFY ELEMENT
NAME IS CUST-CREDIT
COMMENTS NULL.

```

The following example shows how to enter an element definition for an existing element.

```

MODIFY ELEMENT
NAME IS CUST-CREDIT
ELEMENT DEFINITION
- 'CUSTOMER CREDIT RATING. THIS RATING'
- 'IS ASSIGNED TO A CUSTOMER BY THE TREASURER'

```

## Syntax Notation

A uniform system of syntax notation is used throughout the IDMS manuals. The following table explains this system.

TABLE 3-2. SYNTAX NOTATION

NOTATION	MEANING	EXAMPLES
UPPER-CASE WORDS <u>UNDERLINED</u>	keywords required when format is used	<u>FILE DESCRIPTION</u> <i>literal</i>
UPPER-CASE WORDS NOT UNDERLINED	optional keywords used for syntax clarity or completeness of documentation	FREQUENCY IS <i>literal</i>
lower-case words (script)	user text containing any user-supplied data names	NEW NAME IS <i>element name</i>
[ ]	minimum of zero maximum of one	SAME AS FILE <i>file-name</i> [VERSION <i>version-no</i> ]
[ ] ... [ ]	minimum of zero maximum of one each	USER <i>user-name</i> [RESPONSIBLE [FOR [DEFINITION CREATION UPDATE DELETION]]]] ...
[ ] ... [ ]	minimum of one maximum of one each	[INPUT I-O OUTPUT] FILE IS <i>file-name</i> [VERSION <i>version-no</i> ]
{ } 	minimum of one maximum of one each	VALUE IS <i>literal</i> [OCCURS <i>integer</i> TIMES OCCURS 0 TO <i>integer</i> TIMES DEPENDENT ON <i>element-name</i> [INDEXED BY <i>index-name</i> ]]
( )	optional spelling of the immediately preceding word (see note below table)	PICTURE (PIC) IS <i>picture-formatting</i>

NOTE: There is one special operator used in the DDDL syntax which differs from the uniform IDMS syntax notation system, namely (R), the operator for element redefinition. As indicated by the underlining, the parentheses are required characters, i.e., they do not indicate an optional spelling.

## Character Set

In order to afford flexibility in accommodating a broad range of applications, both automated and manual, no restrictions are enforced by the IDD itself with respect to acceptable characters, aside from those associated with these delimiters:

- The period (followed by a blank/space character) is used to terminate a DDDL sentence.
- One or more blanks are used as separators between words and clauses.
- The comma, the semi-colon, and the colon, which are all treated as blanks by the DDDL Compiler, may be used as separators between words and clauses to enhance readability.
- Quotation marks are used to enclose user-supplied names whenever one or more delimiters (blanks, commas, periods, semi-colons, apostrophes, and quote characters) are embedded in the name. For any single site, *either of the two forms of the quote character, but not both*, may be recognized as such when processing DDDL statements. The single quote/apostrophe (') is the default quote character for syntactical purposes. If the double quote character (") is the standard for the user site, the DBL-QUOTE option should be specified at IDMSDIRL execution time.

Inclusion of the alternate form of the quote character, *i.e.*, the form *not* selected as the site standard, in a user-supplied name requires no further syntactical treatment than enclosure of the name in site-standard quotes:

- IDD'S should be input as 'IDD'S' if the site standard is the (default) single quote.
- IDD'S should be input as "IDD'S" if the site standard is the double quote character.

If, however, the site standard form of the quote character is to be embedded in a user-supplied name, it is necessary that the character appear twice for each occurrence in the original name.

- IDD'S should be input as 'IDD' 'S' if the site standard is the (default) single quote.
- IDD'S should be input as "IDD" "S" if the site standard is the double quote character.

IDMS users should bear in mind that IDD entries which are ultimately to be processed by IDMS compilers may consist only of characters in the IDMS character set as specified in the *IDMS Database Design and Definition Guide*.



## Word Types

As indicated above, the DDDL Compiler processes two types of words, 1) keywords and 2) expressions (names, descriptions, other alphanumeric and numeric literals), both user-supplied and system default:

- Keywords are either *required* or *optional*.

They may either be spelled out in full, or they may be abbreviated *ad lib* by eliminating one or more end characters:

ELEMENT  
ELEMEN  
ELEME  
ELEM  
ELE  
EL  
E

Even a single character may be sufficient to unambiguously specify a keyword, if no other keyword for the specific entity category and function that may appear in the same syntactical position in the clause can be identically abbreviated.

Since, for example, both DESCRIPTION and DEFINITION could conceivably be abbreviated as D or DE, neither may be so abbreviated in a sentence for a particular entity category and function if both appear among the keywords for that entity and function and may appear in the same syntactical position.

Optional keywords should also be included in the check for ambiguity: ATTRIBUTES ARE AUTOMATIC may be abbreviated as A A A, but not as A A, since the compiler will identify the second A with the optional ARE. The optional ARE could be unambiguously omitted if the clause were to be abbreviated A AU.

- User-supplied and system default names include system, subsystem, user, file, program, module, element, record, class, and attribute names:
  - If they are to be processed by the IDMS compilers, they may contain only those characters included in the IDMS character set as specified in the *IDMS Database Design and Definition Guide*.
  - If they are to be processed by the IDMS compilers, they should not duplicate any of the IDMS reserved words as specified in the *IDMS Database Design and Definition Guide*.

- If they are ultimately to be processed by higher-level language compilers (COBOL, PL/I, *etc.*), they should not be identical with any of the reserved words of the specific compiler.
- Quotation marks must be used to enclose them whenever one or more delimiters (blanks, commas, periods, semi-colons, apostrophes, and quote characters) are embedded therein.
- Unless a different maximum length is specified in the syntax display for the particular entity category and function, these names must not exceed 32 characters in length.

Additional user-supplied expressions consisting of alphanumeric and numeric literals used to establish initial values for entity/attribute occurrences include:

- Alphanumeric literals which are enclosed in quote characters and can have a maximum length of 34 characters, including the bounding quotes.
- Numeric literals which are *not* enclosed in quotes and can have a maximum length of 31 characters.
- Figurative constants which are reserved words of one or another higher level language compiler (COBOL, PL/I, *etc.*) to be processed as such at program compilation time:

ADD ELEMENT

NAME IS BLANKFIELD

VALUE IS SPACES.

(COBOL reserved word)

DDDL Keywords may be used as user-supplied literals, if desired, as long as the above-cited restrictions are observed:

ADD ELEMENT

NAME IS ELEMENT.

### PREPARED/REVISED BY Clause

To help support the project control function associated with building and maintaining the Dictionary/Directory, the PREPARED BY and REVISED BY clauses permit identification of the individual or project that creates or updates an entity. The identification literal is stored in the Dictionary/Directory along with other information for the entity, and is printed on all detail reports. Literal may not exceed 8 characters.

The format for identification of the individual, or project, who creates an entity with the ADD function follows.

[PREPARED BY *literal*]

The format for identification of the individual, or project, who updates an entity with the MODIFY function follows.

[PREPARED  
REVISED] BY *literal*

The rules for usage with the MODIFY function follow.

- The PREPARED BY option may be designated with the MODIFY function *only if* it was not specified when the entity was added, or during a previous modification.
- The REVISED BY option may always be designated with the MODIFY function. The identification *literal* replaces the code, if any, previously defined for the entity.

### DDDL SENTENCE PRESENTATION SEQUENCE

Sentences may be presented in any sequence, as long as entities referenced in a sentence already exist in the Dictionary/Directory. Within a sentence, clauses may be specified in any order.

- LEVEL NUMBERS are usually established once and for all as the first order of business after the installation process is completed.
- CLASS information is normally entered first in the DDDL input stream.
- ATTRIBUTES are entered next, and reference previously defined classes. Subsequent sentences which define an entity, permit the entity to be associated with attributes defined at this stage.
- SYSTEM information is entered next. Highest level systems (those that are not subordinate to another system) are entered first, as nesting of systems is accomplished through a WITHIN SYSTEM clause. Systems defined at this

time are available for association with users and programs in subsequent sentences.

- SUBSYSTEM functions in every respect as a synonym for SYSTEM.
- USER information is entered next. User definitions established at this time are available for association with programs and records in subsequent sentences.
- FILE information is entered next. File definitions established at this time are available for association with programs and records in subsequent sentences.
- MODULE information is entered next. Modules defined at this time are available for association with programs in subsequent sentences.
- PROGRAM information is entered next. Lowest level programs (those that do not CALL another program) are entered first, as nesting of programs is accomplished through a PROGRAM CALLED clause.
- ELEMENT information is entered next. Lowest level elements (those that do not have subordinate elements) are entered first, as nesting of elements is accomplished through a SUBORDINATE ELEMENTS clause. Elements defined at this time are available for association with records in the RECORD ELEMENT sentence.
- RECORD information is entered last. General record characteristics, such as name and description, are entered in the RECORD sentence, followed by one or more RECORD ELEMENT sentences to associate the record with the appropriate elements.

#### DDDL SENTENCE SYNTAX DISPLAYS

The following discussion covers the format and rules for each DDDL sentence. Sentences are grouped by entity category, and are presented in the sequence suggested above. Where multiple functions may be performed for an entity, the ADD, DELETE, and MODIFY sentences each begin on a new page.

## LEVEL NUMBERS

## LEVEL NUMBERS

### MODIFY LEVEL NUMBERS SENTENCE

The MODIFY LEVEL NUMBERS sentence establishes a standard level number which will be assigned to each record element whose hierarchical position in a record corresponds to the position of the level number coded in this sentence.

**NOTE:** If fewer than 48 level numbers are coded in the MODIFY LEVEL NUMBERS sentence, the default value of 49 will be generated for record elements with unspecified level numbers.

MODIFY LEVEL NUMBERS ARE { *level-number* } ...

- **LEVEL NUMBERS** - *Level-number* must be two digits valued from 02 to 49. Within these limits, any number of *level-numbers* may be specified in ascending order.



## CLASS

## CLASS

### ADD CLASS SENTENCE

The ADD CLASS sentence identifies a class to the Dictionary. Attribute inclusion may be specified as MANUAL or AUTOMATIC, PLURAL or SINGULAR. Comments may be included.

ADD CLASS NAME IS *class-name*

[PREPARED BY *reference*]

[ATTRIBUTES ARE {MANUAL | PLURAL |  
AUTOMATIC | SINGULAR}]

[COMMENTS *comment*]

- CLASS NAME - *Class-name* must be unique. Its length may not exceed 20 characters. If embedded blanks or special characters are included, *class-name* must be enclosed in quotes.
- PREPARED BY - This clause permits the individual responsible for establishing the entity in the Dictionary to log his initials and/or project code. The rules are explained in the earlier discussion of this clause.
- ATTRIBUTES - This clause specifies the two inclusion characteristics for attributes within the class:
  - ATTRIBUTES ARE MANUAL *requires* that attributes within the class be individually predefined via ATTRIBUTE syntax before being associated with a particular dictionary entity occurrence.
  - ATTRIBUTES ARE AUTOMATIC *permits* individual pre-definition of attributes within the class via ATTRIBUTE syntax before association with a particular dictionary entity occurrence. In addition, it provides that attributes may also be included *automatically* within the class when referenced by entity syntax.
  - ATTRIBUTES ARE SINGULAR specifies that a single entity occurrence may be related to *only one* attribute within the associated class.
  - ATTRIBUTES ARE PLURAL specifies that a single entity occurrence may be related to any number of attributes within the associated class.

The default specification is MANUAL PLURAL.

## CLASS

## CLASS

- **COMMENTS** Rules for coding comments are explained in the earlier discussion of the **Coding Format**.

**NOTE:** The following Cullinane-specified classes are added to the Dictionary/Directory as part of the installation process:

Class-name	Included as	Logically associated with
ELEMENT DESIGNATOR	AUTOMATIC PLURAL	ELEMENT
FREQUENCY	MANUAL PLURAL	ELEMENT FILE PROGRAM
LANGUAGE	MANUAL SINGULAR	MODULE PROGRAM
MODE	MANUAL SINGULAR	MODULE PROGRAM RECORD
PRIVACY	MANUAL PLURAL	MODULE
SECURITY	MANUAL PLURAL	ELEMENT FILE MODULE PROGRAM RECORD (SUB)SYSTEM USER

Inasmuch as there is no lockout mechanism involved, users are free to associate the Cullinane-specified classes with any dictionary entity except RECORD ELEMENT.

In like manner, user-specified classes may be associated with any dictionary entity except RECORD ELEMENT.

CLASS

CLASS

### DELETE CLASS SENTENCE

The DELETE CLASS sentence deletes a class from the Dictionary, and all attributes, if any, for this class.

**NOTE:** The DELETE function will be performed even if an attribute for this class is associated with an entity.

DELETE CLASS NAME IS *class-name* .

- **CLASS NAME** - *Class-name* must be a known class.

**NOTE:** The Cullinane-specified classes established at installation time (ELEMENT DESIGNATOR, FREQUENCY, LANGUAGE, MODE, PRIVACY, SECURITY) *cannot* be deleted.

## MODIFY CLASS SENTENCE

The MODIFY CLASS sentence permits a new name and/or new comments to replace the existing ones for a class. Attribute inclusion may be (re)specified as MANUAL or AUTOMATIC, PLURAL or SINGULAR.

MODIFY CLASS NAME IS *class-name*

[PREPARED  
REVISED] BY *literal*

[NEW NAME IS *class-name*]

[ATTRIBUTES ARE | MANUAL | | PLURAL |  
| AUTOMATIC | | SINGULAR |]

[COMMENTS | *comment* |  
| NULL |]

- CLASS NAME - *Class-name* must be a known class.
- PREPARED/REVISED BY - This clause permits the individual responsible for revising the entity to log his initials and/or project code. The rules are explained in the earlier discussion of this clause.
- NEW NAME - This clause renames an existing class entry. *Class-name* must be unique. Its length may not exceed 20 characters. If embedded blanks or special characters are included, *class-name* must be enclosed in quotes.

**NOTE:** The Cullinane-specified classes established at installation time (ELEMENT DESIGNATOR, FREQUENCY, LANGUAGE, MODE, PRIVACY, SECURITY) *cannot* be renamed.

- ATTRIBUTES - This clause (re)specifies the two inclusion characteristics for attributes within the class:
  - ATTRIBUTES ARE MANUAL *requires* that attributes within the class be individually predefined *via* ATTRIBUTE syntax before being associated with a particular dictionary entity occurrence.
  - ATTRIBUTES ARE AUTOMATIC *permits* the individual predefinition of attributes within the class *via* ATTRIBUTE syntax before association with a particular dictionary entity

## CLASS

## CLASS

occurrence. In addition, it provides that attributes may also be included *automatically* within the class when referenced by entity syntax.

- ATTRIBUTES ARE SINGULAR specifies that a single entity occurrence may be related to *only one* attribute within the associated class.
- ATTRIBUTES ARE PLURAL specifies that a single entity occurrence may be related to any number of attributes within the associated class.

The default specification is MANUAL PLURAL.

- COMMENTS - Rules for modifying comments are explained in the earlier discussion of the Coding Format.



## ATTRIBUTE

## ATTRIBUTE

### ADD ATTRIBUTE SENTENCE

The ADD ATTRIBUTE sentence identifies an attribute within a class. Comments may be included.

```

ADD ATTRIBUTE NAME IS
    {
        attribute-name
        security-name
        language-name
        frequency-name
        mode-name
        privacy-name
    }

    WITHIN CLASS
    {
        class-name
        SECURITY
        LANGUAGE
        MODE
        PRIVACY
        FREQUENCY
    }

    [PREPARED BY literal]

    [COMMENTS comment]
  
```

- **ATTRIBUTE NAME** All *attribute-names*, including *security-names*, *language-names*, *mode-names*, *privacy-names*, *frequency-names*, must be unique across all classes. This means, for example, that a *security-name* may not be the same name as a *privacy-name*. A name may not exceed 40 characters. If embedded blanks or special characters are included, the name must be enclosed in quotes.

**NOTE:** In conjunction with the ADDing of the Cullinane-specified classes (ELEMENT DESIGNATOR, FREQUENCY, LANGUAGE, MODE, PRIVACY, and SECURITY), the 'PRODUCTION LOCK' attribute within the class SECURITY is ADDED at installation time. When associated with records and elements, SECURITY IS 'PRODUCTION LOCK' prohibits the DELETE and MODIFY functions; when associated with other dictionary entities, its effect is purely documentational.

- **WITHIN CLASS** - *Class-name* must identify a known class. This clause is required.
- **PREPARED BY** - This clause permits the individual responsible for establishing the entity in the Dictionary to log his initials and/or project code. The rules are explained in the earlier discussion of this clause.
- **COMMENTS** - Rules for coding comments are explained in the earlier discussion of the Coding Format.

# DELETE ATTRIBUTE SENTENCE

The DELETE ATTRIBUTE sentence deletes an attribute from the Dictionary.

**NOTE:** The DELETE function will be performed even if an attribute is associated with an entity.

DELETE ATTRIBUTE NAME IS  $\left\{ \begin{array}{l} \text{attribute-name} \\ \text{entity-name} \\ \text{language-name} \\ \text{mode-name} \\ \text{frequency-name} \\ \text{priority-name} \end{array} \right\}$

- **ATTRIBUTE NAME** - Name must be a known class attribute. Since all attribute-names must be unique, WITHIN CLASS need not be specified.

**NOTE:** The Cullinane-specified attribute 'PRODUCTION LOCK' established at installation time may not be deleted.

## MODIFY ATTRIBUTE SENTENCE

The MODIFY ATTRIBUTE sentence permits a new name and/or comments to replace the existing ones for an attribute.

```

MODIFY ATTRIBUTE NAME IS {
    attribute-name
    security-name
    language-name
    mode-name
    frequency-name
    privacy-name
}

[ PREPARED
  REVISED ] BY {item}

[ NEW NAME IS {
    attribute-name
    security-name
    language-name
    mode-name
    frequency-name
    privacy-name
} ]

[ COMMENTS {comment}
  NULL ]

```

- **ATTRIBUTE NAME** - *Name* must be a known class attribute.
- **PREPARED/REVISED BY** - This clause permits the individual responsible for revising the entity to log his initials and/or project code. The rules are explained in the earlier discussion of this clause.
- **NEW NAME** - All *attribute-names*, including *security-names*, *language-names*, *mode names*, *privacy-names*, and *frequency-names*, must be unique across all classes. A name may not exceed 40 characters. If embedded blanks or special characters are included, the name must be enclosed in quotes.
- **COMMENTS** - Rules for modifying comments are explained in the earlier discussion of the Coding Format.

### ADD [SUB]SYSTEM SENTENCE

The ADD SYSTEM sentence identifies a new system or subsystem to the Dictionary. System descriptions and relationships with other entities may be specified. The reserved word, SUBSYSTEM, may be used interchangeably with SYSTEM. However, a system which is the object of an ADD SUBSYSTEM sentence will not appear on the DDR reports as a SUBSYSTEM unless it is actually declared subordinate to another system by the WITHIN SYSTEM clause.

```
ADD {SUBSYSTEM  
SYSTEM} NAME IS system-name [VERSION IS version-no]
    [PREPARED BY literal]
    [SAME AS {SUBSYSTEM  
SYSTEM} system-name [VERSION version-no]]
    [{SUBSYSTEM  
SYSTEM} DESCRIPTION IS literal]
    [class-name IS attribute-name] ...
    [SECURITY IS security-name] ...
    [WITHIN {SUBSYSTEM  
SYSTEM} system-name [VERSION version-no]] ...
    [COMMENTS comment]
```

- [SUB]SYSTEM NAME - This clause assigns a system its primary name. The concatenation of *system-name* and *version-no* must be unique. *System-name* may not exceed 8 characters. If embedded blanks or special characters are included, *system-name* must be enclosed in quotes. *Version-no* may be one to four digits valued from 1 through 9999. The default is 1. This is the only required clause in the sentence.
- PREPARED BY - This clause permits the individual responsible for establishing this entity in the Dictionary to log his initials and/or project code. The rules are explained in the earlier discussion of this clause.
- SAME AS - The concatenation of *system-name* and *version-no* (default is 1) must identify a known system or subsystem. The descriptions and relationships for this system, including associations with other systems, are copied and assigned the name in the [SUB]SYSTEM NAME clause. Additional clauses modify the copied information through inclusion (multiple-entry clauses) or replacement (single-entry clauses).

## SYSTEM

## SYSTEM

- [SUB]SYSTEM DESCRIPTION - *Literal* is the long or complete name of the system. It may not exceed 40 characters. If embedded blanks or special characters are included, *literal* must be enclosed in quotes.
- *Class-name IS attribute-name* - This clause associates the (sub)system with the designated attribute. *Class-name* must identify a known class.
  - If the inclusion of attributes within the class has been specified, either actively or by default, as MANUAL, *attribute-name* must identify a known attribute within this class which has been previously ADDED *via* ATTRIBUTE syntax. If, however, attribute inclusion for the class has been specified as AUTOMATIC, the named attribute does not need to have been predefined *via* ATTRIBUTE syntax, but will automatically be added to the Dictionary/Directory within the designated class as the result of its specification in this clause.
  - If the default specification of PLURAL is operative, any number of attributes may be associated with the (sub)system *via* the inclusion of multiple attribute clauses. If, however, attributes within the class have been specified as SINGULAR, only one attribute within the class may be associated with any specific (sub)system.
- SECURITY - This clause associates the system with a SECURITY attribute. *Security-name* must identify a known security. Any number of securities may be associated with a system by including multiple SECURITY clauses.
- WITHIN [SUB]SYSTEM - This clause makes the named system subordinate to the WITHIN system. The concatenation of *system-name* and *version-no* (default is 1) must identify a known system or subsystem.
- COMMENTS - Rules for coding comments are explained in the earlier discussion of the Coding Format.



## DELETE [SUB]SYSTEM SENTENCE

The DELETE [SUB]SYSTEM sentence deletes a system or subsystem and all descriptions and relationships associated with it.

DELETE {SUBSYSTEM  
SYSTEM} NAME IS *system-name* [VERSION IS *version-no* ]

- SYSTEM NAME - The concatenation of *system-name* and *version-no* must identify a known system. When *version-no* is not specified, the default is 1.

## MODIFY [SUB]SYSTEM SENTENCE

The MODIFY [SUB]SYSTEM sentence permits system descriptions and relationships to be included, excluded or replaced for a system or subsystem.

```

MODIFY [SUBSYSTEM] NAME IS system-name [VERSION IS version-no]
      [PREPARED/REVISED BY literal]
      [NEW NAME IS system-name]
      [NEW VERSION IS version-no]
      [SUBSYSTEM] DESCRIPTION IS literal
      [INCLUDE/EXCLUDE class-name IS attribute-name] ...
      [INCLUDE/EXCLUDE SECURITY IS security-name] ...
      [INCLUDE/EXCLUDE WITHIN [SUBSYSTEM] system-name [VERSION version-no]] ...
      [COMMENTS {comment} ]
  
```

- [SUB]SYSTEM NAME - The concatenation of *system-name* and *version-no* must identify a known system or subsystem.
- PREPARED/REVISED BY - This clause permits the individual responsible for revising the entity to log his initials and/or project code. The rules are explained in the earlier discussion of this clause.
- NEW NAME - This clause renames an existing system or subsystem. The concatenation of NEW *system-name* and the NEW *version-no* must be unique. When NEW NAME is specified without NEW VERSION, the default is 1. *System-name* may not exceed 8 characters. If embedded blanks or special characters are included, *system-name* must be enclosed in quotes.
- NEW VERSION - This clause assigns a *version-no* to the system specified by the NEW NAME clause. If NEW NAME was not specified, it is a NEW VERSION for the system specified in the [SUB]SYSTEM NAME clause. *Version-no* may be one to four digits valued from 1 through 9999. The default is 1.

- [SUB]SYSTEM DESCRIPTION - The existing description, if any, is replaced by *literal*. *Literal* may not exceed 40 characters. If embedded blanks or special characters are included, *literal* must be enclosed in quotes. To replace an existing description with spaces, code one space enclosed in quotes.
- *class-name* IS *attribute-name* - The EXCLUDE option dissociates the named system and this attribute. If an attribute to be INCLUDED is already associated with the system, a duplicate relationship will not be established. *Class-name* must identify a known class.
  - If the inclusion of attributes within the class has been specified, either actively or by default, as MANUAL, *attribute-name* must identify a known attribute within this class which has been previously ADDED via ATTRIBUTE syntax. If, however, attribute inclusion for the class has been specified as AUTOMATIC, the *named* attribute does not need to have been predefined via ATTRIBUTE syntax, but will automatically be added to the Dictionary/Directory within the designated class as the result of its specification in this clause.
  - If the default specification of PLURAL is operative, any number of attributes may be associated with the (sub)system via the inclusion of multiple attribute clauses. If, however, attributes within the class have been specified as SINGULAR, only one attribute within the class may be associated with any specific (sub)system.
- SECURITY - The EXCLUDE option dissociates the named system and this security. If a security to be INCLUDED is already associated with the system, a duplicate relationship will not be established. Any number of securities may be included or excluded for a system by including multiple SECURITY clauses.
- WITHIN [SUB]SYSTEM - The concatenation of *system-name* and *version-no* must identify a known system or subsystem. The EXCLUDE option dissociates the named system and the WITHIN system. If the system to be INCLUDED is already associated with the named system, a duplicate relationship will not be established. Any number of systems or subsystems may be included or excluded for a system.
- COMMENTS - Rules for modifying comments are explained in the earlier discussion of the Coding Format.

AD-A069 868

NAVAL INTELLIGENCE PROCESSING SYSTEMS SUPPORT ACTIVIT--ETC F/G 9/2  
INTEGRATED DATABASE DEVELOPMENT AND DESIGN GUIDE. VERSION 2.0.(U)  
MAY 79

UNCLASSIFIED

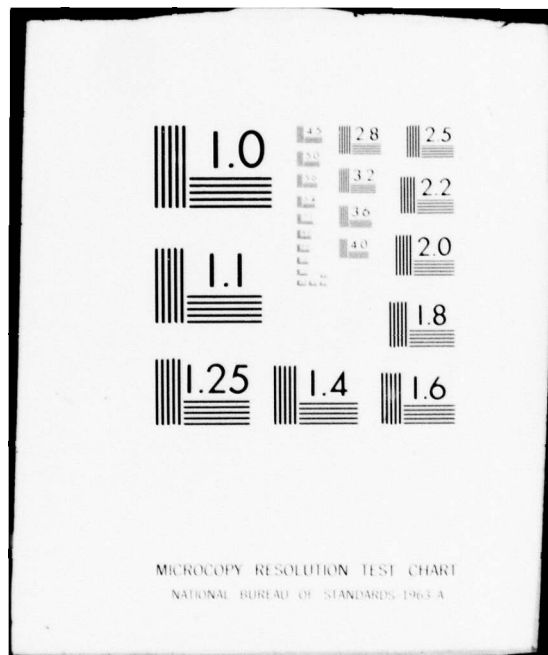
NIPSSA-UM-8000/2.0-5/79

NL

5 OF 7

AD  
A069868







## ADD USER SENTENCE

The ADD USER sentence identifies a user to the Dictionary. User descriptions and relationships with other entities may be specified.

```
ADD USER NAME IS user-name
    [PREPARED BY literal]
    [USER DESCRIPTION IS literal]
    [class-name IS attribute-name] ...
    [SECURITY IS security-name] ...
    [OF (SUBSYSTEM
        SYSTEM) system-name [VERSION version-no]] ...
    [COMMENTS comments]
```

- **USER NAME** - This clause assigns a user its primary name. *User-name* must be unique. Its length may not exceed 32 characters. If embedded blanks or special characters are included, *user-name* must be enclosed in quotes. This is the only required clause in the sentence.
- **PREPARED BY** - This clause permits the individual responsible for establishing the entity in the Dictionary to log his initials and/or project code. The rules are explained in the earlier discussion of this clause.
- **USER DESCRIPTION** - *Literal* is the long or complete name for the user. It may not exceed 40 characters. If embedded blanks or special characters are included, *literal* must be enclosed in quotes.
- ***class-name* IS *attribute-name*** - This clause associates the user with a designated attribute. *Class-name* must identify a known class.
  - If the inclusion of attributes within the class has been specified, either actively or by default, as MANUAL, *attribute-name* must identify a known attribute within this class which has been previously ADDED via ATTRIBUTE syntax. If, however, attribute inclusion for the class has been specified as AUTOMATIC, the named attribute does not need to have been predefined via ATTRIBUTE syntax, but will automatically be added to the Dictionary/Directory within the designated class as the result of its specification in this clause.
  - If the default specification of PLURAL is operative, any number of attributes may be associated with the user via

) the inclusion of multiple attribute clauses. If, however, attributes within the class have been specified as SINGULAR, only one attribute within the class may be associated with any specific user.

- SECURITY - This clause associates the user with a known SECURITY attribute. *Security-name* must be a known security. Any number of securities may be related to the user by including multiple SECURITY clauses.
  - OF [SUB]SYSTEM - This clause associates the user with a system. The concatenation of *system-name* and *version-no* (default is 1) must identify a known system or subsystem. Any number of systems may be associated with the user by including multiple OF [SUB]SYSTEM clauses.
  - COMMENTS - Rules for coding comments are explained in the earlier discussion of the Coding Format.
- )

USER

USER

### DELETE USER SENTENCE

The DELETE USER sentence deletes a user, and all descriptions and relationships associated with it.

DELETE USER NAME IS USER-NAME ;

- **USER-NAME** - *User-name* must identify a known user.

## MODIFY USER SENTENCE

The MODIFY USER sentence permits user descriptions and relationships to be included, excluded, or replaced for a user.

```

MODIFY USER NAME IS user-name

    [ {PREPARED}
      {REVISED} BY literal ]

    [ NEW NAME IS user-name ]

    [ {INCLUDE}
      {EXCLUDE} OF {SUBSYSTEM}
                  system-name [VERSION version-no] ] ...

    [ USER DESCRIPTION IS literal ]

    [ {INCLUDE}
      {EXCLUDE} class-name IS attribute-name ] ...

    [ {INCLUDE}
      {EXCLUDE} SECURITY IS security-name ] ...

    [ COMMENTS {comment}
                {NULL} ]

```

- USER NAME - *User-name* must identify a known user.
- PREPARED/REVISED BY - This clause permits the individual responsible for revising the entity to log his initials and/or project code. The rules are explained in the earlier discussion of this clause.
- NEW NAME - This clause renames an existing user. *User-name* must be unique. Its length may not exceed 32 characters. If embedded blanks or special characters are included, *user-name* must be enclosed in quotes.
- OF [SUB]SYSTEM - The EXCLUDE option dissociates the user and this system. If a system to be INCLUDED is already associated with the user, a new relationship will not be established. Any number of systems or sub-systems may be included or excluded for a user with multiple OF [SUB] SYSTEM clauses.
- USER DESCRIPTION - The existing user description, if any, is replaced by *literal*. *Literal* may not exceed 40 characters. If embedded blanks or special characters are included, *literal* must be enclosed in quotes. To replace an existing description with spaces, code one space enclosed in quotes.
- *class-name* IS *attribute-name* - The EXCLUDE option dissociates the user and this attribute. If an attribute to be INCLUDED is already associated with the user, a duplicate relationship will not be established. *Class-name* must identify a known class.



## USER

## USER

- If the inclusion of attributes within the class has been specified, either actively or by default, as **MANUAL**, *attribute-name* must identify a known attribute within this class which has been previously **ADDED** *via* **ATTRIBUTE** syntax. If, however, attribute inclusion for the class has been specified as **AUTOMATIC**, the named attribute does not need to have been predefined *via* **ATTRIBUTE** syntax, but will automatically be added to the Dictionary/Directory within the designated class as the result of its specification in this clause.
- If the default specification of **PLURAL** is operative, any number of attributes may be associated with the user *via* the inclusion of multiple attribute clauses. If, however, attributes within the class have been specified as **SINGULAR**, only one attribute within the class may be associated with any specific user.
- **SECURITY** - The **EXCLUDE** option dissociates the user and this security. If a security to be **INCLUDED** is already associated with the user, a new relationship will not be established. Any number of securities may be included or excluded for a user with multiple **SECURITY** clauses.
- **COMMENTS** - Rules for modifying comments are explained in the earlier discussion of the **Coding Format**.



## ADD FILE SENTENCE

The ADD FILE sentence identifies a non-IDMS file to the Dictionary. File descriptions and relationships with other entities may be specified. Synonyms (alternate names) may be assigned to a file.

```

ADD FILE NAME IS file-name [VERSION IS version-no]
    [PREPARED BY literal]
    [SAME AS FILE file-name [VERSION version-no]]
    [FILE DESCRIPTION IS literal]
    [EXTERNAL NAME IS external-name]
    [LABELS ARE 

|              |
|--------------|
| STANDARD     |
| NON-STANDARD |
| OMITTED      |

]
    [RELATED FILE IS file-name [VERSION version-no]] ...
    [class-name IS attribute-name] ...
    [SECURITY IS security-name] ...
    [FILE NAME SYNONYM IS file-name] ...
    [COMMENTS comment]

```

- **FILE NAME** - This clause assigns a file its primary name. The concatenation of *file-name* and *version-no* must be unique. *File-name* may not exceed 32 characters. If embedded blanks or special characters are included, *file-name* must be enclosed in quotes. *Version-no* may be one to four digits valued from 1 through 9999. The default is 1. This is the only required clause in the sentence.
- **PREPARED BY** - This clause permits the individual responsible for establishing the entity in the Dictionary to log his initials and/or project code. The rules are explained in the earlier discussion of this clause.
- **SAME AS** - The concatenation of *file-name* and *version-no* (default is 1) must be the primary name for a known file. The descriptions and relationships for this file, including associations with other files, programs, and attributes, are copied and assigned the name in the FILE NAME clause. Synonyms for the file are not copied. Additional clauses modify the copied information through inclusion (multiple-entry clauses) or replacement (single-entry clauses).
- **FILE DESCRIPTION** - *Literal* is the long or complete name of the file. It may not exceed 40 characters. If embedded blanks or special characters are included, *literal* must be enclosed in quotes.

- **EXTERNAL NAME** - *External-name* may not exceed 8 characters. If embedded blanks or special characters are included, *external-name* must be enclosed in quotes.
- **LABELS** - This clause may be used to specify the label processing used for this file. **STANDARD**, **NON-STANDARD** or **OMITTED** may be specified. The default is blanks.
- **RELATED FILE** - This clause associates two files. The concatenation of *file-name* and *version-no* (default is 1) must be the primary name for a known file. Any number of files may be associated with a file by including multiple **RELATED FILE** clauses.
- **class-name IS attribute-name** - This clause associates the file with a designated attribute. *Class-name* must identify a known class.
  - If the inclusion of attributes within the class has been specified, either actively or by default, as **MANUAL**, *attribute-name* must identify a known attribute within this class which has been previously **ADDED** via **ATTRIBUTE** syntax. If, however, attribute inclusion for the class has been specified as **AUTOMATIC**, the named attribute does not need to have been predefined via **ATTRIBUTE** syntax, but will automatically be added to the Dictionary/Directory within the designated class as the result of its specification in this clause.
  - If the default specification of **PLURAL** is operative, any number of attributes may be associated with the file via the inclusion of multiple attribute clauses. If, however, attributes within the class have been specified as **SINGULAR**, only one attribute within the class may be associated with any specific file.
- **SECURITY** - This clause associates the file with a **SECURITY** attribute. *Security-name* must identify a known security. Any number of securities may be associated with a file by including multiple **SECURITY** clauses.
- **FILE NAME SYNONYM** - This clause establishes a synonym for the file. Rules for formation of file synonyms are the same as those for file primary names.
- **COMMENTS** - Rules for coding comments are explained in the earlier discussion of the **Coding Format**.

## FILE

## FILE

### DELETE FILE SENTENCE

The DELETE FILE sentence deletes a file and all descriptions and relationships associated with it.

DELETE FILE NAME IS *file-name* [VERSION IS *version-no*]

- **FILE NAME** - The concatenation of *file-name* and *version-no* must be the primary name for a known file. When *version-no* is omitted, the default is 1.

## MODIFY FILE SENTENCE

The MODIFY FILE sentence permits file descriptions and relationships to be included, excluded or replaced for a file.

```

MODIFY FILE NAME IS file-name [VERSION IS version-no]
    [PREPARED/REVISOR BY initial]
    [NEW NAME IS file-name]
    [NEW VERSION IS version-no]
    [FILE DESCRIPTION IS literal]
    [EXTERNAL NAME IS external-name]
    [LABELS ARE [STANDARD
                  NON-STANDARD
                  OMITTED
                  NULL]]
    [INCLUDE/EXCLUDE RELATED FILE IS file-name [VERSION version-no]] ...
    [INCLUDE/EXCLUDE class-name IS attribute-name]
    [INCLUDE/EXCLUDE SECURITY IS security-name] ...
    [INCLUDE/EXCLUDE FILE NAME SYNONYM IS file-name] ...
    [COMMENTS [COMMENT]
              [NULL]]

```

- **FILE NAME** - The concatenation of *file-name* and *version-no* must be the primary name for a known non-IDMS file.
- **PREPARED/REVISED BY** - This clause permits the individual responsible for revising the entity to log his initials and/or project code. The rules are explained in the earlier discussion of this clause.
- **NEW NAME** - This clause renames an existing file. The concatenation of NEW *file-name* and NEW *version-no* must be unique. When NEW NAME is specified without NEW VERSION, the default is 1. *File-name* may not exceed 32 characters. If embedded blanks or special characters are included, *file-name* must be enclosed in quotes.
- **NEW VERSION** - This clause assigns a *version-no* to the file specified by the NEW NAME clause. If NEW NAME was not specified, it is a NEW VERSION



for the *file-name* specified in the FILE NAME clause. *Version-no* may be one to four digits valued from 1 through 9999. The default is 1.

- FILE DESCRIPTION - The existing description, if any, is replaced by *literal*. *Literal* may not exceed 40 characters. If embedded blanks or special characters are included, *literal* must be enclosed in quotes. To replace an existing description with spaces, code one space enclosed in quotes.
- EXTERNAL NAME - The existing external name, if any, is replaced by *external-name*. *External-name* may not exceed 8 characters. If embedded blanks or special characters are included *external-name* must be enclosed in quotes. To replace an existing external name with spaces, code one space enclosed in quotes.
- LABELS - The option specified replaces the existing label option, if any, for the file.
- RELATED FILE - The EXCLUDE option dissociates the named file and the related file. If a related file to be INCLUDED is already associated with the file, a duplicate relationship will not be established. Any number of related files may be included or excluded for a file with multiple RELATED FILE clauses.
- *class-name* IS *attribute-name* - The EXCLUDE option dissociates the file and this attribute. If an attribute to be included is already associated with a file, a duplicate relationship will not be established. *Class-name* must identify a known class.
  - If the inclusion of attributes within the class has been specified, either actively or by default, as MANUAL, *attribute-name* must identify a known attribute within this class which has been previously ADDED via ATTRIBUTE syntax. If, however, attribute inclusion for the class has been specified as AUTOMATIC, the named attribute does not need to have been predefined via ATTRIBUTE syntax, but will automatically be added to the Dictionary/Directory within the designated class as the result of its specification in this clause.
  - If the default specification of PLURAL is operative, any number of attributes may be associated with the file via the inclusion of multiple attribute clauses. If, however, attributes within the class have been specified as SINGULAR, only one attribute within the class may be associated with any specific file.
- SECURITY - The EXCLUDE option dissociates the file and this security. If a security to be INCLUDED is already associated with the file, a duplicate relationship will not be established. Any number of securities may be included or excluded for a file with multiple SECURITY clauses.



- **FILE NAME SYNONYM** - The **EXCLUDE** option deletes a synonym for the file. The **INCLUDE** option establishes a synonym for a file. All file synonyms must be unique. Rules for the formation of file synonyms are the same as those for primary file names.
- **COMMENTS** - The rules for modifying comments are explained in the earlier discussion of the **Coding Format**.

## ADD MODULE SENTENCE

The ADD MODULE sentence identifies a module to the Dictionary. Module descriptions and relationships with other entities may be specified. Source code for the module may also be catalogued in the Dictionary.

```
ADD MODULE NAME IS module-name [VERSION IS version-no]
    [PREPARED BY license]
    [SAME AS MODULE module-name [VERSION version-no]]
    [MODULE DESCRIPTION IS license]
    [class-name IS attribute-name] ...
    [SECURITY IS security-name] ...
    [PRIVACY IS privacy-name] ...
    [LANGUAGE IS language-name]
    [MODE IS mode-name]
    [COMMENTS comment]
    [MODULE SOURCE FOLLOWS]
        source statements
    MSEND
```

- **MODULE NAME** - This clause assigns a module its primary name. The concatenation of *module-name* and *version-no* must be unique. *Module-name* may not exceed 32 characters. If embedded blanks or special characters are included, *module-name* must be enclosed in quotes. *Version-no* may be one to four digits valued from 1 through 9999. The default is 1. This is the only required clause in the sentence.
- **PREPARED BY** - This clause permits the individual responsible for establishing the entity in the Dictionary to log his initials and/or project code. The rules are explained in the earlier discussion of this clause.
- **SAME AS** - The concatenation of *module-name* and *version-no* (default is 1) must identify a known module. The descriptions and relationships for this module, including associations with programs and attributes, are copied and assigned the name in the MODULE NAME clause. Additional clauses modify the copied information through inclusion (multiple-entry clauses) or replacement (single-entry clauses).

- **MODULE DESCRIPTION** - *Literal* is the long or complete name for the module. *Literal* may not exceed 40 characters. If embedded blanks or special characters are included, *literal* must be enclosed in quotes.
- *class-name IS attribute-name* - This clause associates the module with a designated attribute. *Class-name* must identify a known class.
  - If the inclusion of attributes within the class has been specified, either actively or by default, as **MANUAL**, *attribute-name* must identify a known attribute within this class which has been previously **ADDED** *via* **ATTRIBUTE** syntax. If, however, attribute inclusion for the class has been specified as **AUTOMATIC**, the named attribute does not need to have been predefined *via* **ATTRIBUTE** syntax, but will automatically be added to the Dictionary/Directory within the designated class as the result of its specification in this clause.
  - If the default specification of **PLURAL** is operative, any number of attributes may be associated with the module *via* the inclusion of multiple attribute clauses. If, however, attributes within the class have been specified as **SINGULAR**, only one attribute within the class may be associated with any specific module.
- **SECURITY** - This clause associates the module with a **SECURITY** attribute. *Security-name* must identify a known security. Any number of securities may be associated with the module by including multiple **SECURITY** clauses.
- **PRIVACY** - This clause associates a module with a **PRIVACY** attribute. *Privacy-name* must identify a known privacy. Any number of privacies may be associated with the module by including multiple **PRIVACY** clauses.
- **LANGUAGE** - This clause associates the module with a **LANGUAGE** attribute. *Language-name* must identify a known language. Only one language may be associated with a module.
- **MODE** - This clause associates a module with a **MODE** attribute (**CICS**, **BATCH**, **INTERCOM**, *etc.*). *Mode-name* must identify a known mode. Only one mode may be associated with the module.

**NOTE:** The mode designated for a module can affect the **COPY** function performed by the **COBOL** and **PL/I** processors.

- **COMMENTS** - Rules for coding comments are explained in the earlier discussion of the **Coding Format**.
- **MODULE SOURCE** - *Source-statement* is the actual source code in 80-character card format. **DML** commands will be properly intercepted by

MODULE

MODULE

the COBOL and PL/I processors and translated into CALL statements. COPY functions will be performed by the processors even when included within the copied module source (nested copies).

**DELETE MODULE SENTENCE**

The DELETE MODULE sentence deletes a module, all descriptions and relationships associated with it, and the source code catalogued for the module.

DELETE MODULE NAME IS *module-name* [VERSION IS *version-no*] ;

- **MODULE NAME** - The concatenation of *module-name* and *version-no* must identify a known module. When *version-no* is not specified, the default is 1.



## MODIFY MODULE SENTENCE

The MODIFY MODULE sentence permits descriptions and relationships to be included, excluded or replaced for a module. Source code which has been catalogued for the module may be replaced.

```
MODIFY MODULE NAME IS module-name [VERSION IS version-no]
```

```
  [PREPARED  
  REVISED] BY literal
```

```
  [NEW NAME IS module-name]
```

```
  [NEW VERSION IS version-no]
```

```
  [MODULE DESCRIPTION IS literal]
```

```
  [INCLUDE  
  EXCLUDE] class-name IS attribute-name ...
```

```
  [INCLUDE  
  EXCLUDE] SECURITY IS security-name ...
```

```
  [INCLUDE  
  EXCLUDE] PRIVACY IS privacy-name ...
```

```
  [LANGUAGE IS {language-name  
  NULL}]
```

```
  [INCLUDE  
  EXCLUDE] MODE IS mode-name
```

```
  [COMMENTS {comment  
  NULL}]
```

```
  [MODULE SOURCE FOLLOWS  
  source statements  
  MSEND]
```

- **MODULE NAME** - The concatenation of *module-name* and *version-no* (default is 1) must identify a known module.
- **PREPARED/REVISED BY** - This clause permits the individual responsible for revising the entity to log his initials and/or project code. The rules are explained in the earlier discussion of this clause.
- **NEW NAME** - This clause renames an existing module entry. The concatenation of NEW *module-name* and NEW *version-no* must be unique. When NEW NAME is specified without NEW VERSION, the default is 1. *Module-name* may not exceed 32 characters. If embedded blanks or special characters are included, *module-name* must be enclosed in quotes.

## MODULE

## MODULE

- **NEW VERSION** - This clause assigns a new *version-no* to the module specified by the **NEW NAME** clause. If **NEW NAME** is not specified, it is a **NEW VERSION** for the module specified in the **MODULE NAME** clause. *Version-no* may be one to four digits valued from 1 through 9999. The default is 1.
- **MODULE DESCRIPTION** - The existing description, if any, is replaced by *literal*. *Literal* may not exceed 40 characters. If embedded blanks or special characters are included, *literal* must be enclosed in quotes. To replace an existing description with spaces, code one space enclosed in quotes.
- *class-name* **IS** *attribute-name* - The **EXCLUDE** option dissociates the module and this attribute. If an attribute to be **INCLUDED** is already associated with the module, a duplicate relationship will not be established. *Class-name* must identify a known class.
  - If the inclusion of attributes within the class has been specified, either actively or by default, as **MANUAL**, *attribute-name* must identify a known attribute within this class which has been previously **ADDED** *via* **ATTRIBUTE** syntax. If, however, attribute inclusion for the class has been specified as **AUTOMATIC**, the named attribute does not need to have been predefined *via* **ATTRIBUTE** syntax, but will automatically be added to the Dictionary/Directory within the designated class as the result of its specification in this clause.
  - If the default specification of **PLURAL** is operative, any number of attributes may be associated with the module *via* the inclusion of multiple attribute clauses. If, however, attributes within the class have been specified as **SINGULAR**, only one attribute within the class may be associated with any specific module.
- **SECURITY** - The **EXCLUDE** option dissociates the module and this security. If a security to be **INCLUDED** is already associated with the module, a duplicate relationship will be established. Any number of securities may be included or excluded for a module.
- **PRIVACY** - The **EXCLUDE** option dissociates the module and this privacy. If a privacy to be **INCLUDED** is already associated with the module, a duplicate relationship will not be established. Any number of privacies may be included or excluded for a module.
- **LANGUAGE** - The existing language name, if any, is replaced by *language-name* or deleted by **NULL**.
- **MODE** - The **EXCLUDE** option dissociates the module and this mode. If a mode to be **INCLUDED** is already associated with the module, a duplicate relationship will not be established. Only one mode may be included for a module.

## MODULE

## MODULE

- **COMMENTS** - Rules for modifying comments are explained in the earlier discussion of the **Coding Format**.
- **MODULE SOURCE** - The existing source code for the module, if any, is replaced by *source statements*. All source statements for a module must be replaced as a unit.

## PROGRAM

## PROGRAM

### ADD PROGRAM SENTENCE

The ADD PROGRAM sentence identifies a program to the Dictionary. Program descriptions and relationships with other entities may be specified.

When the COBOL and PL/I processors examine a source program, they log their statistics and other information into the Dictionary on the basis of *program-name*. The processors will supplement program data already entered in the Dictionary if the *program-name* established in this sentence agrees with the program source.

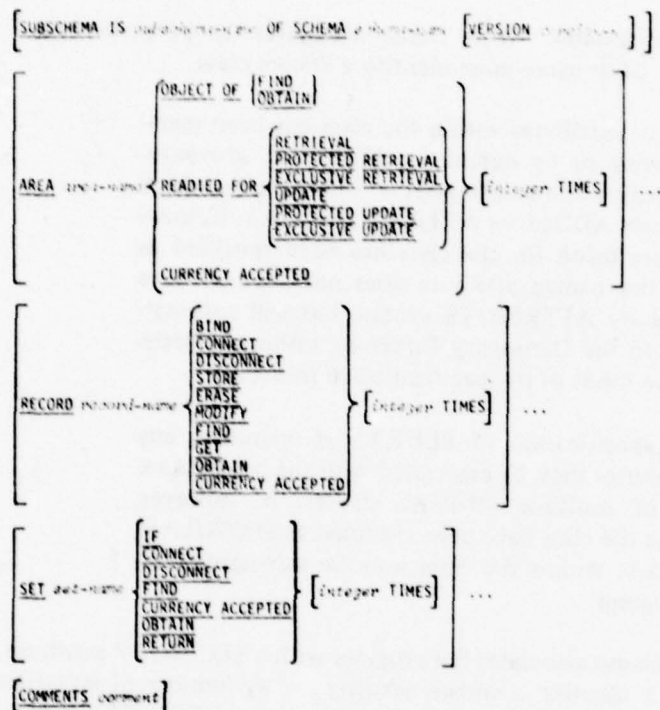
The processors require that the file name used in the OPEN statement be a primary name or synonym for a file in order to log the program's file usage. When the file has been previously defined to the Dictionary, the processor recognizes each OPEN for the *file-name*. On the first OPEN for a usage, (INPUT/I-O/OUTPUT), the processor creates a program-file record, related to the program and to the file, which contains a usage code and a usage count of 1. Subsequent OPENS for the same *file-name* with the same usage increment a usage counter for that program-file combination.

```
ADD PROGRAM NAME IS program-name [VERSION IS version-no]
    [PREPARED BY literal]
    [SAME AS PROGRAM program-name [VERSION version-no]]
    [PROGRAM DESCRIPTION IS literal]
    [class-name IS attribute-name] ...
    [SECURITY IS security-name] ...
    [LANGUAGE IS language-name]
    [MODE IS mode-name]
    [WITHIN (SUBSYSTEM
             SYSTEM) system-name [VERSION version-no]] ...
    [ESTIMATED LINES ARE integer]
    [PROGRAM CALLED IS {program-name [VERSION version-no]} ...] ...
    [MODULE USED IS {module-name [VERSION version-no]} ...] ...
    [INPUT
     I-O
     OUTPUT] FILE IS file-name [VERSION version-no] [integer TIMES] ...
```



# PROGRAM

# PROGRAM



- **PROGRAM NAME** - This clause assigns a program its primary name. The concatenation of *program-name* and *version-no* must be unique. *Program-name* may not exceed 8 characters. If embedded blanks or special characters are included, *program-name* must be enclosed in quotes. *Version-no* may be one to four digits valued from 1 through 9999. The default is 1. This is the only required clause in the sentence.
- **PREPARED BY** - This clause permits the individual responsible for establishing the entity in the Dictionary to log his initials and/or project code. The rules are explained in the earlier discussion of this clause.
- **SAME AS** - The concatenation of *program-name* and *version-no* (default is 1) must identify a known program. The descriptions and relationships for this program, including associations with called programs, files, records, modules, and attributes, are copied and assigned the name in the PROGRAM NAME clause. Additional clauses modify the copied information through inclusion (multiple-entry clauses) or replacement (single-entry clauses).
- **PROGRAM DESCRIPTION** - *Literal* is the long or complete name for the program. It may not exceed 40 characters. If embedded blanks or special characters are included, *literal* must be enclosed in quotes.



## PROGRAM

- *class-name* IS *attribute-name* - This clause associates the program with a designated attribute. *Class-name* must identify a known class.
- If the inclusion of attributes within the class has been specified, either actively or by default, as MANUAL, *attribute-name* must identify a known attribute within this class which has been previously ADDED via ATTRIBUTE syntax. If, however, attribute inclusion for the class has been specified as AUTOMATIC, the named attribute does not need to have been predefined via ATTRIBUTE syntax, but will automatically be added to the Dictionary/Directory within the designated class as the result of its specification in this clause.
- If the default specification of PLURAL is operative, any number of attributes may be associated with the program via the inclusion of multiple attribute clauses. If, however, attributes within the class have been specified as SINGULAR, only one attribute within the class may be associated with any specific program.
- SECURITY - This clause associates the program with a SECURITY attribute. *Security-name* must identify a known security. Any number of securities may be associated with the program by including multiple SECURITY clauses.
- LANGUAGE - This clause associates the program with a LANGUAGE attribute. *Language-name* must identify a known language. Only one language may be associated with a program.
- MODE - This clause associates the program with a MODE attribute. *Mode-name* must identify a known mode. Only one mode may be associated with the program.
- WITHIN [SUB]SYSTEM - This clause associates the program with a system. The concatenation of *system-name* and *version-no* (default is 1) must identify a known system or subsystem. Any number of systems may be associated with the program by including multiple [SUB]SYSTEM clauses.

**NOTE:** The information accepted by the following eight clauses is automatically logged into the Dictionary for each program processed by the COBOL and PL/I processors. In cases where another entity is referenced, both the DDDL compiler and the processors require that the referenced entity be present in the Dictionary.

- ESTIMATED LINES - This clause accepts the estimated number of lines, or cards, of source code for the program. *Integer* may be 1 to 16 digits.
- PROGRAM CALLED - This clause associates the program with another program. The concatenation of *program-name* and *version-no* (defaults to 1)

## PROGRAM

## PROGRAM

## PROGRAM

must identify a known program. Any number of called programs may be associated with a program by including multiple PROGRAM CALLED clauses, or by coding multiple *program-names*.

- **MODULE USED** - This clause associates the program with a module. The concatenation of *module-name* and *version-no* (default is 1) must identify a known module. Any number of modules may be associated with the program by including multiple MODULE USED clauses, or by coding multiple *modules-names*.

**NOTE:** The program-to-module relationship established at this time causes the module to appear on the Program Report as a COPYMOD with the DICTIONARY flag, and causes the program to appear on the Module Report as a PROGRAM with the DICTIONARY flag.

- **INPUT/I-O/OUTPUT FILE** - This clause logs statistics for the program's usage of a non-IDMS file. The concatenation of *file-name* and *version-no* must be the primary name or a synonym for a known file. *Integer* is the usage count, or number of times the file is OPENED with this usage. Any number of file usages may be associated with a program by including multiple INPUT/I-O/OUTPUT FILE clauses.
- **SUBSCHEMA** - *Subschema-name* must be the name of a known subschema. The concatenation of *schema-name* and *version-no* must identify a schema related to the subschema.
- **AREA** - This clause logs statistics for the program's usage of an area. *Area-name* must identify an area in the named subschema. *Integer*, a user-supplied count of the usage of one of the listed IDMS functions, may be one to four digits valued from 1 through 9999. The default is 1.
- **RECORD** - This clause logs statistics for the program's usage of an IDMS record. *Record-name* must identify a record in the named subschema. *Integer*, a user-supplied count of the usage of one of the listed IDMS functions, may be one to four digits valued from 1 through 9999. The default is 1.
- **SET** - This clause logs statistics for the program's usage of a set. *Set-name* must identify a set in the named subschema. *Integer*, a user-supplied count of the usage of one of the listed IDMS functions, may be one to four digits valued from 1 through 9999. The default is 1.
- **COMMENTS** - Rules for coding comments are explained in the earlier discussion of the Coding Format.

## PROGRAM

## PROGRAM

### DELETE PROGRAM SENTENCE

The DELETE PROGRAM sentence deletes a program and all descriptions and relationships associated with it.

DELETE PROGRAM NAME IS *program-name* [VERSION IS *version-no*]

- **PROGRAM NAME** - The concatenation of *program-name* and *version-no* must identify a known program. When *version-no* is not specified, the default is 1.

# PROGRAM

# PROGRAM

## MODIFY PROGRAM SENTENCE

The MODIFY PROGRAM sentence permits descriptions and relationships to be included, excluded or replaced for a program.

```

MODIFY PROGRAM NAME IS program-name [VERSION IS version-no]

    [PREPARED  
REVISED] BY literal

    [NEW NAME IS program-name]

    [NEW VERSION IS version-no]

    [PROGRAM DESCRIPTION IS literal]

    [INCLUDE  
EXCLUDE] alias-name IS attribute-name ...

    [INCLUDE  
EXCLUDE] SECURITY IS security-name ...

    [LANGUAGE IS {language-name  
NULL}]

    [MODE IS {mode-name  
NULL}]

    [INCLUDE  
EXCLUDE] WITHIN {SUBSYSTEM  
SYSTEM} system-name [VERSION version-no] ...

    [ESTIMATED LINES ARE integer]

    [INCLUDE  
EXCLUDE] PROGRAM CALLED IS {program-name [VERSION version-no] } ...

    [INCLUDE  
EXCLUDE] MODULE USED IS {module-name [VERSION version-no] } ...

    [INCLUDE  
EXCLUDE] {INPUT  
I-O  
OUTPUT} FILE IS file-name [VERSION version-no] [integer TIMES] ...

    [INCLUDE  
EXCLUDE] SUBSCHEMA IS subschema-name OF SCHEMA schema-name [VERSION version-no]

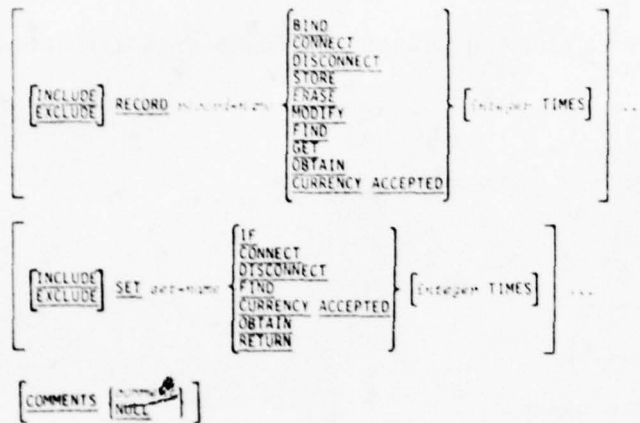
    [INCLUDE  
EXCLUDE] AREA area-name {
        OBJECT OF {FIND  
OBTAIN}
        READIED FOR {RETRIEVAL RETRIEVAL  
PROTECTED RETRIEVAL  
EXCLUSIVE RETRIEVAL  
UPDATE  
PROTECTED UPDATE  
EXCLUSIVE UPDATE}
        CURRENCY ACCEPTED
    } {integer TIMES} ...

```



## PROGRAM

## PROGRAM



- **PROGRAM NAME** - The concatenation of *program-name* and *version-no* (default is 1) must identify a known program.
- **PREPARED/REVISED BY** - This clause permits the individual responsible for revising the entity to log his initials and/or project code. The rules are explained in the earlier discussion of this clause.
- **NEW NAME** - This clause renames an existing program. The concatenation of *NEW program-name* and *NEW version-no* must be unique. When *NEW NAME* is specified without *NEW VERSION*, the default is 1. *Program-name* may not exceed 8 characters. If embedded blanks or special characters are included, *program-name* must be enclosed in quotes.
- **NEW VERSION** - This clause assigns a *version-no* to the program specified by the *NEW NAME* clause. If *NEW NAME* is not specified, it is a *NEW VERSION* for the *program-name* specified in the *PROGRAM NAME* clause. *Version-no* may be one to four digits valued from 1 through 9999. The default is 1.
- **PROGRAM DESCRIPTION** - The previously specified description, if any, is replaced by *literal*. *Literal* may not exceed 40 characters. If embedded blanks or special characters are included, *literal* must be enclosed in quotes. To replace an existing description with spaces, code one space enclosed in quotes.
- *class-name IS attribute-name* - The *EXCLUDE* option dissociates the program and this attribute. If an attribute to be *INCLUDED* is already associated with the system, a duplicate relationship will not be established. *Class-name* must identify a known class.



- If the inclusion of attributes within the class has been specified, either actively or by default, as *MANUAL*, *attribute-name* must identify a known attribute within this class which has been previously ADDED *via* *ATTRIBUTE* syntax. If, however, attribute inclusion for the class has been specified as *AUTOMATIC*, the named attribute does not need to have been predefined *via* *ATTRIBUTE* syntax, but will automatically be added to the Dictionary/Directory within the designated class as the result of its specification in this clause.
- If the default specification of *PLURAL* is operative, any number of attributes may be associated with the program *via* the inclusion of multiple attribute clauses. If, however, attributes within the class have been specified as *SINGULAR*, only one attribute within the class may be associated with any specific program.
- *SECURITY* - The *EXCLUDE* option dissociates the program and this security. If a security to be *INCLUDED* is already associated with the program, a new relationship will not be established. Any number of securities may be included or excluded from a program with multiple *SECURITY* clauses.
- *LANGUAGE* - The existing language, if any, is replaced by *language-name* or deleted by *NULL*. *Language-name* must be a known *LANGUAGE* attribute.
- *MODE* - The existing mode, if any, is replaced by *mode-name* or deleted by *NULL*.

**NOTE:** The COBOL and PL/I processors assign default *mode-names* to programs not associated with a mode. The default mode assigned to COBOL programs is *DMLC BATCH*; for PL/I programs it is *DMLP BATCH*.

- *WITHIN [SUB]SYSTEM* - The *EXCLUDE* option dissociates the program and the *WITHIN* system. If the system to be *INCLUDED* is already associated with the program, a duplicate relationship will not be established. Any number of systems or subsystems may be included or excluded for a program *via* multiple *WITHIN [SUB]SYSTEM* clauses.
- *ESTIMATED LINES* - The previously specified count, if any, is replaced by *integer*. It may be one to 16 digits.
- *PROGRAM CALLED* - The *EXCLUDE* option dissociates the program and the called program. If a program to be *INCLUDED* is already associated with the program, a duplicate relationship will not be established. Any number of called programs may be included or excluded for a program with multiple *PROGRAM CALLED* clauses, or by coding multiple *program-names*.

- **MODULE USED** - The EXCLUDE option dissociates the program and this module. If a module to be INCLUDED is already associated with the program, a duplicate relationship will not be established. Any number of modules may be included or excluded for a program by including multiple MODULE USED clauses or by coding multiple *module-names*.
- **INPUT/I-O/OUTPUT FILE** - The EXCLUDE option dissociates the program and this file usage. If a file usage to be INCLUDED is already associated with the program, a duplicate relationship will not be established.
  - **TIMES** - To alter the usage count for a file usage, first EXCLUDE INPUT/I-O/OUTPUT FILE *file-name*, then INCLUDE INPUT/I-O/OUTPUT FILE *file-name integer* with the new usage count.
- **SUBSCHEMA** - Only one subschema may be related to the program at a time. To modify the name of a program's subschema, first specify EXCLUDE SUBSCHEMA using the *subschema-name*, *schema-name*, and *version-no* (default is 1) of the previously specified subschema. (Then designate INCLUDE SUBSCHEMA using the *subschema-name*, *schema-name*, and *version-no* (default is 1) of the new subschema.) If the subschema is EXCLUDED, all area, record and set statistics are deleted along with the subschema reference.
- **AREA** - The exclude option dissociates the program and this area, either for selective usages when OBJECT OF, READIED FOR or CURRENCY ACCEPTED are specified, or for all usages, when only *area-name* is specified. To INCLUDE an area usage, follow the rules in the ADD PROGRAM sentence.
- **RECORD** - The EXCLUDE option dissociates the program and this record, either for selective usages, when IDMS functions are specified, or for all usages, when only *record-name* is specified. To INCLUDE a record usage, follow the rules in the ADD PROGRAM sentence.
- **SET** - The EXCLUDE option dissociates the named program and this set, either for selective usages, when IDMS functions are specified, or for all usages, when only *set-name* is specified. To INCLUDE a set usage, follow the rules in the ADD PROGRAM sentence.
- **COMMENTS** - Rules for modifying comments are explained in the earlier discussion of the Coding Format.

## ADD ELEMENT SENTENCE

The ADD ELEMENT sentence identifies an element to the Dictionary. Element descriptions and relationships with other entities may be specified. Synonyms (alternate names) can be assigned to an element. Condition names and elements designated as subordinate to a group element must have been previously defined.

ADD ELEMENT NAME IS *element-name*

[PREPARED BY *list-name*]

[SAME AS ELEMENT *element-name*]

[ELEMENT DESCRIPTION IS *list-name*]

[ELEMENT DEFINITION IS *definition*]

[ELEMENT DESIGNATOR IS *designator-name*] ...

[SUBORDINATE ELEMENTS ARE { { *element-name* [OCCURS *integer*] } ... }  
NULL ]

[PICTURE IS *picture-name*] ...

[USAGE IS {  
BIT  
POINTER  
CONDITION-NAME  
DISPLAY  
COMP(BINARY)  
COMP-1(SHORT-POINT)  
COMP-2(LONG-POINT)  
COMP-3(PACKED)  
}]

[VALUE IS *initial-value*]

[JUSTIFY IS {ON  
OFF}]

[BLANK WHEN ZERO IS {ON  
OFF}]

[*list-name* IS *initial-value*] ...

[SECURITY IS *security-name*] ...

[FREQUENCY IS *frequency-name*]

[USER IS *user-name* [RESPONSIBLE [FOR {  
DEFINITION  
CREATION  
UPDATE  
DELETION  
}]]] ...

[ELEMENT NAME SYNONYM IS *synonym-name* [FOR GROUP SYNONYM *group-name*]] ...

[RANGE IS *range-name* [THRU *range-name*]] ...

[COMMENTS *comments*]

## ELEMENT

The rules for the ADD ELEMENT sentence follow.

- **ELEMENT NAME** - This clause assigns an element its primary name. *Element-name* must be unique. Its length may not exceed 32 characters. If embedded blanks or special characters are included, *element-name* must be enclosed in quotes. This is the only required clause in the sentence.
- **PREPARED BY** - This clause permits the individual responsible for establishing the entity in the Dictionary to log his initials and/or project code. The rules are explained in the earlier discussion of this clause.
- **SAME AS** - *Element-name* must identify a known element. The descriptions and relationships for this element, including associations with users, subordinate elements, and attributes, are copied and assigned the name in the ELEMENT NAME clause. Synonyms and record elements are not copied. Additional clauses modify the copied information through inclusion (multiple-entry clauses) or replacement (single-entry clauses).
- **ELEMENT DESCRIPTION** - *Literal* is the long or complete name of the element. It may not exceed 64 characters. If embedded blanks or special characters are included, *literal* must be enclosed in quotes.
- **ELEMENT DEFINITION** - This clause permits the definition of an element to be separated from other remarks or comments. Rules for coding the definition are explained in the discussion of general rules above.
- **ELEMENT DESIGNATOR** - This clause may be used to identify elements with a common characteristic which is peculiar to elements. ELEMENT DESIGNATOR is a class. *Designator-name* automatically becomes an attribute within the ELEMENT DESIGNATOR class, associated with this element. Rules for the formation of *designator-name* are the same as those for attributes.
- **SUBORDINATE ELEMENTS** - This clause identifies the element as a group element and identifies those elements which constitute the group. *Element-name* is the primary name for a known element (elementary or group). Multiple subordinate elements may be submitted on multiple cards without the continuation character.
  - **OCCURS** - *Integer* is the number of times the element occurs. If it is a variably occurring element, specify the maximum.
  - **(R)** - Subordinate elements may be redefined. The *element-name* of the element to be redefined is entered first, followed by the *element-name* of the element which does the redefining. The required operation (R) is the third component.



**NOTE:** The symbol for this operator consists of three characters: a left parenthesis, a capital R, and a right parenthesis. The parentheses do *not* indicate an optional abbreviation.

**NOTE:** To define a filler as a subordinate element, specify an element name of 'FIL $mm$ ' where  $mm$  is the number of characters of filler. For example, to generate a filler of FILLER PIC X(7), code the following subordinate element: 'FIL 0007'.

**NOTE:** The five clauses which follow define the element's physical characteristics.

- **PICTURE** - This clause identifies the element as an elementary item. It specifies the length and data type (numeric, alpha, etc.). *Character-string* may not exceed 32 characters. If embedded blanks or special characters are included, *character-string* must be enclosed in quotes.
- **USAGE** - This clause designates the data usage for the element. When usage is **CONDITION-NAME**, a level number of 88 will be generated for this element. The default is **DISPLAY**. **USAGE** options follow.

USAGE option	Meaning	Acceptable to this processor
BIT	Bit string definition	IDMSDMLP
POINTER	Fullword Address Constant	IDMSDMLP
CONDITION-NAME	Level 88 values	IDMSDMLC
COMP (BINARY)	Binary	IDMSDMLC and IDMSDMLP
COMP-1 (SHORT-POINT)	Short precision floating point	IDMSDMLC and IDMSDMLP
COMP-2 (LONG-POINT)	Long precision floating point	IDMSDMLC and IDMSDMLP
COMP-3 (PACKED)	Packed decimal	IDMSDMLC and IDMSDMLP
DISPLAY	Zoned decimal	IDMSDMLC and IDMSDMLP

- **VALUE** - The COBOL and PL/I processors sometimes initialize an element with the value specified in the Dictionary. A value may be specified whether or not an element is assigned a picture. *Initial-value* may not exceed 32 characters. If embedded blanks or special characters are included, *initial-value* must be enclosed in quotes.
- **JUSTIFY** - This clause may be used to specify non-standard justification of a data element. Specify **ON** to justify numeric fields left or alphanumeric fields right. The default is **OFF**.
- **BLANK WHEN ZERO** - Specify **ON** to cause zero suppression if the element is all zeros. The default is **OFF**.
- *class-name* **IS** *attribute-name* - This clause associates the element with a designated attribute. *Class-name* must identify a known class.



- If the inclusion of attributes within the class has been specified, either actively or by default, as **MANUAL**, *attribute-name* must identify a known attribute within this class which has been previously **ADDED** via **ATTRIBUTE** syntax. If, however, attribute inclusion for the class has been specified as **AUTOMATIC**, the named attribute does not need to have been predefined via **ATTRIBUTE** syntax, but will automatically be added to the Dictionary/Directory within the designated class as the result of its specification in this clause.
- If the default specification of **PLURAL** is operative, any number of attributes may be associated with the element via the inclusion of multiple attribute clauses. If, however, attributes within the class have been specified as **SINGULAR**, only one attribute within the class may be associated with any specific element.
- **SECURITY** - This clause associates the element with a **SECURITY** attribute. *Security-name* must identify a known security. Any number of securities may be associated with an element by including multiple **SECURITY** clauses.

**NOTE:** The **DELETE** function and most **MODIFY** functions may be prohibited by including the following clause:

**SECURITY IS 'PRODUCTION LOCK'**

- **FREQUENCY** - This clause associates the element with a **FREQUENCY** attribute. *Frequency-name* must identify a known frequency. Only one frequency may be associated with an element.
- **USER** - This clause associates the element with a user. *User-name* must identify a known user. Any number of users may be associated with an element by including multiple **USER** clauses.
- **RESPONSIBLE FOR DEFINITION CREATION UPDATE DELETION** - A user's responsibility for an element may be specified by including one or more of the **FOR** options. If the **FOR** options are not specified, but **RESPONSIBLE** is present, the user is assumed to be responsible for all functions. If only the user name is specified, the user is assumed to have only access to the element.
- **ELEMENT NAME SYNONYM** - The clause assigns a synonym (alternate name) to an element. Rules for the formation of synonym names are the same as those for primary names, except that duplicate synonym names are permitted.

- **ELEMENT NAME SYNONYM FOR GROUP SYNONYM** - This clause assigns a synonym to a subordinate element and 'associates' it with the synonym for a group element. This is the only exception to the backward referencing rule, since neither the group element nor its synonym can be known to the Dictionary at this time. Rules for the formation of synonym names are the same as those for primary names, except that duplicate synonym names *are* permitted.
- **RANGE** - This clause may be used to document the acceptable values for an element. These values may be specified by entering each acceptable value using one or more RANGE clauses, or by entering ranges of acceptable values using one or more RANGE...THRU clauses. *Literal* may not exceed 32 characters. If embedded blanks or special characters are included, *literal* must be enclosed in quotes.
- **COMMENTS** - Rules for coding comments are explained in the earlier discussion of the Coding Format.

## ELEMENT

## ELEMENT

### DELETE ELEMENT SENTENCE

The DELETE ELEMENT sentence deletes an element and all descriptions and relationships associated with it.

**NOTE:** The DELETE function will not be performed if an element is associated with a record, if an element is subordinate in a group structure, if it is in 'PRODUCTION LOCK' status, or if it was created by the Schema Compiler or by the IDMSCLUC utility.

DELETE ELEMENT NAME IS *element-name* ..

- **ELEMENT NAME** - *Element-name* must be the primary name for a known element.

**NOTE:** When group elements are deleted, any fillers which were part of the group structure are *not* deleted from the Dictionary. Fillers may be removed by submitting a DELETE ELEMENT sentence with the name used to create the filler.

When the PREFIX/SUFFIX option is specified for a record synonym, an element synonym is automatically generated for each element subsequently associated with that record synonym. (The synonym is composed of the prefix or suffix *literal* and the *element-name*.) This element synonym can be deleted by submitting the following sentence:

#### MODIFY ELEMENT

NAME IS *element-name*

EXCLUDE ELEMENT NAME SYNONYM IS  
*concatenated element-name*

## MODIFY ELEMENT SENTENCE

The MODIFY ELEMENT sentence permits descriptions and relationships to be included, excluded or replaced for an element.

MODIFY ELEMENT NAME IS *element-name*

[PREPARED  
REVISED] BY *literal*

[NEW NAME IS *element-name*]

[ELEMENT DESCRIPTION IS *literal*]

[ELEMENT DEFINITION IS { *definition*  
NULL }]

[INCLUDE  
EXCLUDE] ELEMENT DESIGNATOR IS *designator-name* ...

[SUBORDINATE ELEMENTS ARE { { *element-name* [OCCURS *integer*]  
(*in buffer for element-name* and *multiple designator-name* (R)) ... }  
NULL }]

[PICTURE *picture-definition*]

[USAGE IS {  
BIT  
POINT  
CONDITION-NAME  
DISPLAY  
COMP(BINARY)  
COMP-1(SHORT-POINT)  
COMP-2(LONG-POINT)  
COMP-3(PACKED)  
}]

[VALUE IS { *initial-value*  
NULL }]

[JUSTIFY IS { ON  
OFF }]

[BLANK WHEN ZERO IS { ON  
OFF }]

[INCLUDE  
EXCLUDE] *alias-name* IS *attribute-name* ...

[INCLUDE  
EXCLUDE] SECURITY IS *security-name* ...

[FREQUENCY IS { *frequency-name*  
NULL }]

[INCLUDE  
EXCLUDE] USER IS *user-name* [RESPONSIBLE [FOR { DEFINITION  
CREATION  
UPDATE  
DELETION } ]]] ...

```

[ [INCLUDE  
EXCLUDE] ELEMENT NAME SYNONYM IS literal [FOR GROUP SYNONYM literal] ] ...
[ [INCLUDE  
EXCLUDE] RANGE IS literal [THRU literal] ] ...
[ COMMENTS [ literal ] ]

```

- **ELEMENT NAME** - *Element-name* must be the primary name for a known element.
- **PREPARED/REVISED BY** - This clause permits the individual responsible for revising the entity to log his initials and/or project code. The rules are explained in the earlier discussion of this clause.
- **NEW NAME** - This clause renames an existing element entry. *Element-name* must be unique. Its length may not exceed 32 characters. If embedded blanks or special characters are included, *element-name* must be enclosed in quotes.
- **ELEMENT DESCRIPTION** - The existing description, if any, is replaced by *literal*. *Literal* may not exceed 40 characters. If embedded blanks or special characters are included, *literal* must be enclosed in quotes. To replace an existing description with spaces, code one space enclosed in quotes.
- **ELEMENT DEFINITION** - Rules for modifying the element definition are explained in the discussion of general rules above. The existing definition, if any, is replaced by *definition* or deleted by NULL.
- **ELEMENT DESIGNATOR** - The EXCLUDE option dissociates the element and this element designator. For the INCLUDE option, if *designator-name* is an existing element designator, the element is associated with it. Otherwise, *designator-name* automatically becomes an attribute within the ELEMENT DESIGNATOR class, associated with this element.
- **SUBORDINATE ELEMENTS** - The first *element-name* encountered dissociates all previously specified subordinate elements and the group item identified by the ELEMENT NAME clause. This means that all subordinate elements must be replaced as a unit. Follow the rules in the ADD ELEMENT sentence. The NULL option dissociates subordinate elements and the group item.
- **OCCURS** - To alter the number of times an element occurs, all subordinate elements must be resubmitted along with the changed *integer*.



## ELEMENT

## ELEMENT

- (R) - Subordinate elements may be redefined. The *element-name* of the element to be redefined is entered first, followed by the *element-name* of the element which does the redefining. The required operator (R) is the third component.

**NOTE:** The symbol for this operator consists of three characters: a left parenthesis, a capital R, and a right parenthesis. The parentheses do *not* indicate an optional abbreviation.

**NOTE:** The five clauses which follow replace the physical characteristics previously specified for the element. Follow the rules in the ADD ELEMENT sentence.

- PICTURE
- USAGE
- VALUE
- JUSTIFY
- BLANK WHEN ZERO
- *classname* IS *attribute-name* - The EXCLUDE option dissociates the element and this attribute. If an attribute to be INCLUDED is already associated with the element, a duplicate relationship will not be established. *Class-name* must identify a known class.
- If the inclusion of attributes within the class has been specified, either actively or by default, as MANUAL, *attribute-name* must identify a known attribute within this class which has been previously ADDED *via* ATTRIBUTE syntax. If, however, attribute inclusion for the class has been specified as AUTOMATIC, the named attribute does not need to have been predefined *via* ATTRIBUTE syntax, but will automatically be added to the Dictionary/Directory within the designated class as the result of its specification in this clause.
- If the default specification of PLURAL is operative, any number of attributes may be associated with the element *via* the inclusion of multiple attribute clauses. If, however, attributes within the class have been specified as SINGULAR, only one attribute within the class may be associated with any specific element.

## ELEMENT

- **SECURITY** - The EXCLUDE option dissociates the element and this security. If a security to be INCLUDED is already associated with the element a new relationship will not be established. Any number of securities may be included or excluded for an element.

**NOTE:** The MODIFY function will not be performed if the SECURITY attribute 'PRODUCTION LOCK' is associated with the element.

- **FREQUENCY** - A previously specified FREQUENCY, if any, is replaced by *literal* or deleted by NULL. *Literal* may not exceed 40 characters. If embedded blanks or special characters are included, *literal* must be enclosed in quotes.
- **USER** - The EXCLUDE option dissociates the element and this user responsibility. The INCLUDE option specifies a user's responsibility for the element. To designate a user with more than one responsibility, INCLUDE the user multiple times with different RESPONSIBLE FOR options.
- **ELEMENT NAME SYNONYM** - The EXCLUDE option deletes a synonym for the named element. The INCLUDE option assigns a synonym to the element. Rules for the formation of synonym names are the same as those for primary names, except that duplicate synonym names are permitted.
- **ELEMENT NAME SYNONYM FOR GROUP SYNONYM** - The EXCLUDE option dissociates the subordinate element synonym and the group synonym. The INCLUDE option assigns a synonym to the subordinate element and associates it with the synonym for a group element.

**NOTE:** If an element synonym-to-group synonym relationship has been established, but the EXCLUDE does not specify FOR GROUP SYNONYM, the specified synonym for the subordinate element will be deleted, whether or not it has an associated group synonym.

- **RANGE** - The EXCLUDE option deletes an acceptable value or value range as previously specified. The INCLUDE option identifies a new acceptable value or value range. Follow the rules in the ADD ELEMENT sentence.
- **COMMENTS** - Rules for modifying comments are explained in the earlier discussion of the Coding Format.

## ELEMENT

## ADD RECORD SENTENCE RECORD ELEMENT SUBSENTENCE

The ADD RECORD sentence identifies a record to the Dictionary. Record descriptions and relationships with other entities may be specified. A record may be associated with a file. The order in which the records are available within a file and/or the direct-access key for a record within a file may be specified. Synonyms (alternate names) may be assigned to a record and, if appropriate, the synonym may be associated with a file name, or with a file name synonym.

When elements which participate in a record are to have a name which is composed of an element synonym plus a prefix or suffix, the value of the prefix or suffix may be assigned at this time.

The RECORD ELEMENT subsentence associates an element with the immediately preceding ADD RECORD sentence. Group elements, elementary elements not contained in a group, and fillers may be designated as record elements. When a prefix or suffix has been defined and associated with a record synonym for the record, a concatenated element name is generated at this time and associated with the record synonym.

```

ADD RECORD NAME IS record-name [VERSION IS version-no]
    [PREPARED BY literal]
    [SAME AS RECORD record-name [VERSION version-no]]
    [RECORD DESCRIPTION IS literal]
    [RECORD STORAGE IS [AUTOMATED]
                        [MANUAL]
                        literal]
    [OCCURRENCES ARE integer]
    [class-name IS attribute-name] ...
    [SECURITY IS security-name] ...
    [MODE IS mode-name]
    [WITHIN FILE file-name [VERSION version-no] [ [KEY IS element-name [ASCENDING]
                                                    [DESCENDING] ] ... ] ] ...
    [RECORD NAME SYNONYM FOR FILE [SYNONYM] file-name IS record-name [VERSION version-no] [ [PREFIX]
                                                    [SUFFIX] literal ] ] ...
    [RECORD NAME SYNONYM IS record-name [VERSION version-no] [ [PREFIX]
                                                    [SUFFIX] literal ] ] ...
    [COMMENTS comment]
  
```

# RECORD/RECORD ELEMENT

# RECORD/RECORD ELEMENT

RECORD ELEMENT IS *element-name*

[ELEMENT NAME SYNONYM [FOR RECORD SYNONYM *record-name* [VERSION *version-name*]] IS *element-name* ] ...

[PICTURE [IS *character-string*]]

[USAGE IS {  
BIT  
POINTER  
CONDITION-NAME  
DISPLAY  
COMP (BINARY)  
COMP-1 (SHORT-POINT)  
COMP-2 (LONG-POINT)  
COMP-3 (PACKED)  
}]

[REDEFINES *element-name*]

[OCCURS *integer* TIMES

[OCCURS 0 TO *integer* TIMES DEPENDING ON *element-name*]

[INDEXED BY *index-name*]

[INDEX KEY IS *element-name* [ASCENDING  
DESCENDING]]

[SYNC]

[COMMENTS {*comment*;  
NULL }]

SUBORDINATE ELEMENT IS *element-name*

[PICTURE [IS *character-string*]]

[USAGE IS {  
BIT  
POINTER  
CONDITION-NAME  
DISPLAY  
COMP (BINARY)  
COMP-1 (SHORT-POINT)  
COMP-2 (LONG-POINT)  
COMP-3 (PACKED)  
}]

[OCCURS *integer* TIMES

[OCCURS 0 TO *integer* TIMES DEPENDING ON *element-name*]

[INDEXED BY *index-name*]

[INDEX KEY IS *element-name* [ASCENDING  
DESCENDING]]

[SYNC]

[COMMENTS {*comment*;  
NULL }]



- RECORD NAME - This clause assigns a record its primary name. The concatenation of *record-name* and *version-no* must be unique. *Record-name* may not exceed 32 characters. If embedded blanks or special characters are included, *record-name* must be enclosed in quotes. *Version-no* may be one to four digits valued from 1 to 9999. The default is 1. This is the only required clause in the sentence.
- PREPARED BY - This clause permits the individual responsible for establishing the entity in the Dictionary to log his initials and/or project code. The rules are explained in the earlier discussion of this clause.
- SAME AS - The concatenation of *record-name* and *version-no* (default is 1) must identify a known record. The descriptions and relationships for this record are copied and assigned the name in the RECORD NAME clause. Synonyms and record elements specified for the record are *not* copied. Additional clauses modify the copied information through inclusion (multiple-entry clauses) or replacement (single-entry clauses).
- RECORD DESCRIPTION - *Literal* is the long or complete name of the record. It may not exceed 40 characters. If embedded blanks or special characters are included, *literal* must be enclosed in quotes.
- RECORD STORAGE - This clause specifies the storage medium which is used for this record. *Literal* may not exceed 16 characters. If embedded blanks or special characters are included, *literal* must be enclosed in quotes.
- OCCURRENCES - This clause indicates the number of occurrences for this record. *Integer* may be 1 to 16 digits.
- *class-name* IS *attribute-name* - This clause associates the record with a designated attribute. *Class-name* must identify a known class.
  - If the inclusion of attributes within the class has been specified, either actively or by default, as MANUAL, *attribute-name* must identify a known attribute within this class which has been previously ADDED *via* ATTRIBUTE syntax. If, however, attribute inclusion for the class has been specified as AUTOMATIC, the named attribute does not need to have been predefined *via* ATTRIBUTE syntax, but will automatically be added to the Dictionary/Directory within the designated class as the result of its specification in this clause.
  - If the default specification of PLURAL is operative, any number of attributes may be associated with the record *via* the inclusion of multiple attribute clauses. If, however, attributes within the class have been specified as SINGULAR, only one attribute within the class may be associated with any specific record.



## RECORD/RECORD ELEMENT

## RECORD/RECORD ELEMENT

- **SECURITY** - This clause associates the record with a SECURITY attribute. *Security-name* must identify a known security. Any number of securities may be associated with a record by including multiple SECURITY clauses.

**NOTE:** The DELETE function and the MODIFY function may be prohibited by including the following clause:

SECURITY IS 'PRODUCTION LOCK'

- **MODE** - This clause associates the record with a MODE attribute. *Mode-name* must be a known mode. Only one mode may be associated with a record by including multiple MODE clauses.

**NOTE:** The mode designated for a record can affect the COPY function performed by the COBOL and PL/I processors.

- **WITHIN FILE** - This clause associates the record with a designated file. The concatenation of *file-name* and *version-no* must be the primary name for a known file. Any number of files may be associated with a record by including multiple WITHIN FILE clauses.
- **KEY** - Keys for a file may be specified by including one or more key clauses. A maximum of five keys may be specified. A key is an element in a record which is used to locate a record (direct-access key), or an element which identifies the ordering of records within a file. ASCENDING or DESCENDING defines the sequence of sorted records. *Element-name* is the primary name for a known element.

**NOTE:** The following clause has four functions. It associates a record synonym with a designated file or file synonym, establishes a synonym for the record if it does not exist, and can designate a prefix or suffix which will be attached to all elements associated with this record synonym.

- **RECORD NAME SYNONYM FOR FILE** - This option associates the record synonym with a file. *File-name* must be the primary name for a known file. If the concatenation of *record-name* and *version-no* does not identify a synonym for this record, the record synonym is established at this time.
- **RECORD NAME SYNONYM FOR FILE SYNONYM** - This option associates the record synonym with a file synonym. *File-name* must be the synonym for a known file. If the concatenation of *record-name* and *version-no* does not identify a synonym for this record, the record synonym is established at this time.

- The PREFIX/SUFFIX option designates a literal which will be attached to the name of elements associated with this record synonym. The literal will *not* be attached to the record name. The concatenated element name may not exceed 32 characters. *Literal* may not exceed 6 characters.
  - PREFIX causes the literal to be concatenated on the front of the element name. (The dash, if desired, should be the last character of the literal.)
  - SUFFIX causes the literal to be concatenated on the end of the element name. (The dash, if desired, should be the first character of the literal).
- RECORD NAME SYNONYM - This clause assigns a synonym (alternate name) to a record. The concatenation of *record-name* and *version-no* (default is 1) is the synonym name for the record. Rules for the formation of synonym names are the same as those for record primary names. This clause is used when no relationship should be established between the record name synonym and a file.
- COMMENTS - Rules for coding comments are explained in the earlier discussion of the **Coding Format**.
- RECORD ELEMENT - This clause associates an element with the preceding RECORD sentence. *Element-name* must be the primary name for a known element, or it must identify the element as a filler. The relationship between this element name and all record names (primary and synonyms) is established at this time. If the prefixed or suffixed name to be associated with the record is the primary element name plus the prefix or suffix, the concatenated element name is established automatically by the system at this time, along with the relationship.

**NOTE:** To define a filler as a record element, specify an element name of 'FIL *nnnn*', where *nnnn* is the number of characters of filler. For example, to generate a filler of FILLER PIC X(7), code the following record element: 'FIL 0007'.

- ELEMENT NAME SYNONYM - This clause associates an element synonym and the primary record name. If *element-name* is not a known synonym for this element, it is established at this time.
- ELEMENT NAME SYNONYM FOR RECORD SYNONYM...IS - This option associates an element synonym and a record synonym. If the *element-name* is not a known synonym for this element, it is established at this time. If the prefixed or suffixed name to be associated with this record synonym is this element synonym plus the prefix or suffix, the concatenated element name is established at this time, along with the relationship. The concatenation of *record-name* and *version-no* must be a known synonym for this record.

# RECORD/RECORD ELEMENT

# RECORD/RECORD ELEMENT

- **PICTURE** - This clause is used only for group elements. When it is not present, subordinate elements will be automatically included in the record at this time. When it is present, the subordinate elements will not be included. Instead, an alphanumeric display picture will be included for the group element. Its length will be calculated automatically.
- **PICTURE IS** - This clause may be used for either group elements or primary (lowest level) elements. For group elements, the subordinate elements will not be included. Instead, the specified *character-string* will be included as the picture for the group element. For primary elements, the specified *character-string* will replace any previously specified picture.
- **USAGE** - This clause (re)designates the data usage for the element. When usage is **CONDITION-NAME**, a level number of 88 will be generated for this element. The default is **DISPLAY**. **USAGE** options follow.

USAGE option	Meaning	Acceptable to this processor
BIT	Bit string definition	IDMSDMLP
POINTER	Fullword Address Constant	IDMSDMLP
CONDITION-NAME	Level 88 values	IDMSDMLC
COMP (BINARY)	Binary	IDMSDMLC and IDMSDMLP
COMP-1 (SHORT-POINT)	Short precision floating point	IDMSDMLC and IDMSDMLP
COMP-2 (LONG-POINT)	Long precision floating point	IDMSDMLC and IDMSDMLP
COMP-3 (PACKED)	Packed decimal	IDMSDMLC and IDMSDMLP
DISPLAY	Zoned decimal	IDMSDMLC and IDMSDMLP

- **REDEFINES** - This clause designates an element to redefine the preceding record element. *Element-name* must be the primary name for a known element.
- **OCCURS** - When the record element occurs a fixed number of times within this record, the **OCCURS** clause must specify the number of times it occurs. *Integer* may be one to four digits valued from 1 through 9999.
- **OCCURS DEPENDING ON** - When the record element occurs a variable number of times within this record, this clause must specify the maximum number of times it may occur. *Integer* may be one to four digits valued from 1 through 9999. *Element-name* must be a primary element name.
- **INDEXED BY** - This clause may be specified only when a record element is the subject of an **OCCURS** or **OCCURS...DEPENDING ON** clause. *Index-name* is stored away in the Dictionary and is inserted into a program's data division code by the COBOL processor as part of the record **COPY** function.
- **INDEX KEY** - This clause specifies the ordering of a multiply occurring element which is subordinate to this element. *Element-name* must be a primary element name. **ASCENDING** or **DESCENDING** defines the sequence of the ordered elements.

- SYNC - This clause will cause boundary alignment for the record element when it is copied into a source program according to the following rules:

If element USAGE is...	RECORD ELEMENT will be aligned on...
POINTER	FULLWORD
COMP (BINARY)	HALFWORD if S9(4) or less, else FULLWORD
COMP-1 (SHORT-POINT)	FULLWORD
COMP-2 (LONG-POINT)	DOUBLEWORD

- COMMENTS - Rules for coding comments are explained in the earlier discussion of the Coding Format.

**NOTE:** The following clauses are *not* used to define subordinate elements for this RECORD ELEMENT. The subordinate element-to-group element relationship is established in the ELEMENT sentence for the group element. The following clauses may be used to specify further length characteristics for multiply occurring subordinate elements, to specify alignment for the subordinate element, and to document the usage of index keys for tables (arrays).

- SUBORDINATE ELEMENT - A subordinate element-to-group element relationship must have been established between this element and the group element named in the RECORD ELEMENT clause. *Element-name* must be the primary element name. Subordinate record elements must be named in separate SUBORDINATE ELEMENT clauses, and must be input to the same sequence as they were coded in the SUBORDINATE ELEMENT clause in the ELEMENT sentence for the group.

**NOTE:** The OCCURS *integer* specified in the following clauses overrides the OCCURS *integer* clauses, if any, previously specified in the ELEMENT sentence.

- PICTURE - This clause is used only for group elements. When it is not present, subordinate elements will be automatically included in the record at this time. When it is present, the subordinate elements will not be included. Instead, an alphanumeric display picture will be included for the group element. Its length will be calculated automatically.
- PICTURE IS - This clause may be used for either group elements or primary (lowest level) elements. For group elements, the subordinate elements will not be included. Instead, the specified *character-string* will be included as the picture for the group element. For primary elements, the specified *character-string* will replace any previously specified picture.
- USAGE - This clause (re)designates the data usage for the element. When usage is CONDITION-NAME, a level number of 88 will be generated for this element. The default is DISPLAY. USAGE options follow.



## RECORD/RECORD ELEMENT

## RECORD/RECORD ELEMENT

USAGE option	Meaning	Acceptable to this processor
BIT	Bit string definition	IDMSDMLP
POINTER	Fullword Address Constant	IDMSDMLP
CONDITION-NAME	Level 88 values	IDMSDMLC
COMP (BINARY)	Binary	IDMSDMLC and IDMSDMLP
COMP-1 (SHORT-POINT)	Short precision floating point	IDMSDMLC and IDMSDMLP
COMP-2 (LONG-POINT)	Long precision floating point	IDMSDMLC and IDMSDMLP
COMP-3 (PACKED)	Packed decimal	IDMSDMLC and IDMSDMLP
DISPLAY	Zoned decimal	IDMSDMLC and IDMSDMLP

- OCCURS - When the subordinate element is repeated a fixed number of times within this group element, this clause specifies the number of times it may occur. *Integer* may be one to four digits valued from 1 through 9999.
- OCCURS DEPENDING ON - When the subordinate element is repeated a variable number of times within this group element, this clause specifies the maximum number of times it may occur along with the DEPENDING ON *element-name*. *Integer* may be one to four digits valued from 1 through 9999. *Element-name* must be a primary element name.
- INDEXED BY - This clause may be specified only when a subordinate element is the subject of an OCCURS or OCCURS...DEPENDING ON clause. *Index-name* is stored away in the Dictionary and is inserted into a program's data division code by the COBOL processor as part of the record COPY function.
- INDEX KEY - This clause specifies the ordering of a multiply occurring element which is subordinate to this element. *Element-name* must be a primary element name. ASCENDING or DESCENDING defines the sequence of the ordered elements.
- SYNC - This clause will cause boundary alignment for the subordinate element when it is copied into a source program according to the following rules:

If element USAGE is...	SUBORDINATE RECORD ELEMENT will be aligned on...
POINTER	FULLWORD
COMP (BINARY)	HALFWORD if S9(4) or less, else FULLWORD
COMP-1 (SHORT-POINT)	FULLWORD
COMP-2 (LONG-POINT)	DOUBLEWORD

- COMMENTS - Rules for coding comments are explained in the earlier discussion of the Coding Format.



## DELETE RECORD SENTENCE

The DELETE RECORD sentence deletes a record and all descriptions and relationships associated with it, including the record element relationships. (The elements themselves, as freestanding dictionary entities, are *not* deleted.)

**NOTE:** The DELETE function will not be performed if a record is associated with the 'PRODUCTION LOCK' attribute of the SECURITY class, or if it was created by the Schema Compiler or the IDMSCLUC utility.

Individual record elements are removed by means of the REMOVE RECORD ELEMENT subsub-sentence, which is subordinate to the REBUILD RECORD ELEMENTS subsentence of the MODIFY RECORD sentence.

DELETE RECORD NAME IS *record-name* [VERSION IS *version-no*]

- **RECORD NAME** - The concatenation of *record-name* and *version-no* must be the primary name for a known record. When *version-no* is not specified, the default is 1.

RECORD/RECORD ELEMENT

RECORD/RECORD ELEMENT

MODIFY RECORD SENTENCE

RECORD ELEMENT SUBSENTENCE

REBUILD RECORD ELEMENTS SUBSENTENCE

RECORD ELEMENT SUBSUBSENTENCE

REMOVE RECORD ELEMENT SUBSUBSENTENCE

The MODIFY RECORD sentence permits descriptions and relationships to be included, excluded, or replaced for a record.

- The inclusion of a RECORD ELEMENT subsentence immediately following the MODIFY RECORD sentence (*i.e.*, without an intervening REBUILD RECORD ELEMENTS subsentence) dissociates the record from *all* previously specified record elements, requiring that *all* desired record elements be reentered, even if only one of the record elements was to be changed.
- The REBUILD RECORD ELEMENTS subsentence (submitted instead of the RECORD ELEMENT subsentence) permits both *implicit* and *explicit* modification of individual record elements. *Implicitly*, modifications made to the (freestanding) elements *via* MODIFY ELEMENT syntax will be effected in the corresponding individual record elements if the REBUILD RECORD ELEMENTS subsentence is submitted. Individual record elements may be *explicitly* modified by means of subsequent RECORD ELEMENT subsentences, meanwhile preserving the remaining previously built record elements intact. RECORD ELEMENT clauses must be submitted in the sequence in which the elements occur. New elements may be included at the end of the record.
- The specification of schema name and schema version in this sentence permits *limited descriptive* modification of schema records. Neither structural modification nor removal of schema record elements is permitted.
- The submission of the REMOVE RECORD ELEMENT subsentence following the REBUILD RECORD ELEMENTS subsentence permits the *explicit* removal of individual record elements. (The elements themselves, as freestanding dictionary entities, are *not* deleted thereby.)
- Non-IDMS record element modification and removal may be *explicitly* effected using the REBUILD RECORD ELEMENTS sentence in conjunction with the RECORD ELEMENT and REMOVE RECORD ELEMENT subsentences.

# RECORD/RECORD ELEMENT

# RECORD/RECORD ELEMENT

MODIFY RECORD NAME IS *record-name* [ (VERSION IS *version-number*)  
 (OF SCHEMA NAME IS *schema-name*) (VERSION IS *version-number*) ] ]

[ (PREPARED  
 REVISED ) BY *person* ]

[ NEW NAME IS *new-name* ]

[ NEW VERSION IS *version-number* ]

[ RECORD DESCRIPTION IS *description* ]

[ RECORD STORAGE IS (AUTOMATED  
 MANUAL ) ]

[ OCCURRENCES ARE *integer* ]

[ (INCLUDE  
 EXCLUDE ) *statement* IS *statement* ] ...

[ (INCLUDE  
 EXCLUDE ) SECURITY IS *security-name* ] ...

[ (INCLUDE  
 EXCLUDE ) MODE IS *mode-name* ] ...

[ (INCLUDE  
 EXCLUDE ) WITHIN FILE *file-name* (VERSION *version-number*) [ (KEY IS *element-name* (ASCENDING  
 DESCENDING ) ) ... ] ] ...

[ (INCLUDE  
 EXCLUDE ) RECORD NAME SYNONYM FOR FILE [SYNONYM] *file-name* IS *statement* (VERSION *version-number*) [ (PREFIX  
 SUFFIX ) *literal* ] ] ...

[ (INCLUDE  
 EXCLUDE ) RECORD NAME SYNONYM IS *statement* (VERSION *version-number*) [ (PREFIX  
 SUFFIX ) *literal* ] ] ...

[ COMMENTS { *comment* } ]

# RECORD/RECORD ELEMENT

# RECORD/RECORD ELEMENT

RECORD ELEMENT IS *element-name*

[ELEMENT NAME SYNONYM [FOR RECORD SYNONYM *record-name* [VERSION *version*]] IS *element-name* ]

[PICTURE [IS *picture-name*]]

[USAGE IS {  
 BIT  
 POINTER  
 CONDITION-NAME  
 DISPLAY  
 COMP (BINARY)  
 COMP-1 (SHORT-POINT)  
 COMP-2 (LONG-POINT)  
 COMP-3 (PACKED)  
 }]

[REDEFINES *element-name*]

[OCCURS *integer* TIMES

[OCCURS 0 TO *integer* TIMES DEPENDING ON *element-name*]

[INDEXED BY *index-name*]

[INDEX KEY IS *element-name* [ASCENDING  
 DESCENDING]]

[SYNC]

[COMMENTS {  
 NULL  
 }]

SUBORDINATE ELEMENT IS *element-name*

[PICTURE [IS *picture-name*]]

[USAGE IS {  
 BIT  
 POINTER  
 CONDITION-NAME  
 DISPLAY  
 COMP (BINARY)  
 COMP-1 (SHORT-POINT)  
 COMP-2 (LONG-POINT)  
 COMP-3 (PACKED)  
 }]

[OCCURS *integer* TIMES

[OCCURS 0 TO *integer* TIMES DEPENDING ON *element-name*]

[INDEXED BY *index-name*]

[INDEX KEY IS *element-name* [ASCENDING  
 DESCENDING]]

[SYNC]

[COMMENTS {  
 NULL  
 }]

[REBUILD RECORD ELEMENTS.]

# RECORD/RECORD ELEMENT

# RECORD/RECORD ELEMENT

RECORD ELEMENT IS *element-name*

[ELEMENT NAME SYNONYM [FOR RECORD SYNONYM *name* [VERSION *number*] ] IS *name* ] ...

[PICTURE [IS *data-representation* ]]

[USAGE IS {  
 BIT  
 POINTER  
 CONDITION-NAME  
 DISPLAY  
 COMP (BINARY)  
 COMP-1 (SHORT-POINT)  
 COMP-2 (LONG-POINT)  
 COMP-3 (PACKED)  
 } ]

[REDEFINES *element-name* ]

[OCCURS *integer* TIMES  
 [OCCURS 0 TO *integer* TIMES DEPENDING ON *element-name* ]

[INDEXED BY *index-name* ]

[INDEX KEY IS *element-name* [ASCENDING  
DESCENDING ]]

[SYNC ]

[COMMENTS { *comment* }  
 [ NULL ] ]

SUBORDINATE ELEMENT IS *element-name*

[PICTURE [IS *data-representation* ]]

[USAGE IS {  
 BIT  
 POINTER  
 CONDITION-NAME  
 DISPLAY  
 COMP (BINARY)  
 COMP-1 (SHORT-POINT)  
 COMP-2 (LONG-POINT)  
 COMP-3 (PACKED)  
 } ]

[OCCURS *integer* TIMES  
 [OCCURS 0 TO *integer* TIMES DEPENDING ON *element-name* ]

[INDEXED BY *index-name* ]

[INDEX KEY IS *element-name* [ASCENDING  
DESCENDING ]]

[SYNC ]

[COMMENTS { *comment* }  
 [ NULL ] ]

[REMOVE RECORD ELEMENT NAME IS *element-name* ] ...



## RECORD/RECORD ELEMENT

## RECORD/RECORD ELEMENT

- RECORD NAME

- VERSION - The concatenation of *record-name* and *record-version-no* must be the primary name for a known record. *Record-version-no* may consist of from one to four digits ranging in value from 1 to 9999. The default value is 1.
- SCHEMA NAME - The concatenation of *schema-name* and *schema-version-no* must identify a schema in which the record participates. *Schema-version-no* may consist of from one to four digits ranging in value from 1 to 9999. If *schema-version-no* is not specified, VERSION will default to the highest *schema-version-no* found for this schema. Schema record modification may be effected via the following clauses:

PREPARED BY  
REVISED BY  
RECORD DESCRIPTION  
RECORD STORAGE  
OCCURRENCES  
ATTRIBUTES  
RECORD SYNONYM NAMES  
COMMENTS

Limited *explicit* modification of schema record elements via the REBUILD RECORD ELEMENTS and RECORD ELEMENT clauses may be effected by means of the following clauses:

ELEMENT NAME SYNONYM  
COMMENTS

- PREPARED/REVISED BY - This clause permits the individual responsible for revising the entity to log his initials and/or project code. The rules are explained in the discussion of general rules above.
- NEW NAME - This clause renames an existing record entry. The concatenation of NEW *record-name* and NEW *version-no* must be unique. When NEW NAME is specified without NEW VERSION, the default is 1. *Record-name* may not exceed 32 characters. If embedded blanks or special characters are included, *record-name* must be enclosed in quotes.
- NEW VERSION - This clause assigns a *version-no* to the *record-name* specified by the NEW NAME clause. If NEW NAME was not specified, it is a NEW VERSION for the *record-name* specified in the RECORD NAME clause. *Version-no* may be one to four digits valued from 1 through 9999. The default is 1.

- RECORD DESCRIPTION - The existing description, if any, is replaced by *literal*. *Literal* may not exceed 40 characters. If embedded blanks or special characters are included, *literal* must be enclosed in quotes. To replace an existing description with spaces, code one space enclosed in quotes.
- RECORD STORAGE - This clause specifies the storage medium which is used for this record. *Literal* may not exceed 16 characters. If embedded blanks or special characters are included, *literal* must be enclosed in quotes.
- OCCURRENCES - This clause indicates the number of occurrences for this record. *Integer* may be 1 to 16 digits.
- *class-name* IS *attribute-name* - The EXCLUDE option dissociates the named record and this attribute. If an attribute to be INCLUDED is already associated with the record, a new relationship will not be established. *Class-name* must identify a known class.
  - If the inclusion of attributes within the class has been specified, either actively or by default, as MANUAL, *attribute-name* must identify a known attribute within this class which has been previously ADDED via ATTRIBUTE syntax. If, however, attribute inclusion for the class has been specified as AUTOMATIC, the named attribute does not need to have been predefined via ATTRIBUTE syntax, but will automatically be added to the Dictionary/Directory within the designated class as the result of its specification in this clause.
  - If the default specification of PLURAL is operative, any number of attributes may be associated with the record via the inclusion of multiple attribute clauses. If, however, attributes within the class have been specified as SINGULAR, only one attribute within the class may be associated with any specific record.
- SECURITY - The EXCLUDE option dissociates the record and this security. If a security to be INCLUDED is already associated with the record, a duplicate relationship will not be established. Any number of securities may be included or excluded for a record.

**NOTE:** To remove a record from 'PRODUCTION LOCK' status, submit the following sentence:

EXCLUDE SECURITY IS 'PRODUCTION LOCK'.

- MODE - The EXCLUDE option dissociates the record and this mode. If a mode to be INCLUDED is already associated with the record, a new relationship will not be established.

- **WITHIN FILE** - The EXCLUDE option dissociates the record and this file. If a file to be INCLUDED is already associated with the record, a duplicate relationship will not be established.
- **KEY** - Key fields (SORT and CALC keys) are maintained for file/record relationships by means of this clause. *Element-name* is the primary name for a known element. ASCENDING or DESCENDING defines the sequence of sorted records. If the file/record relationship was previously defined, the key fields are modified. If the relationship was not previously defined, the relationship and the key fields will be established.
- **RECORD NAME SYNONYM FOR FILE** - The EXCLUDE option dissociates this record and the designated file. *File-name* must be the primary name for a file which is associated with this record synonym. For the INCLUDE option, rules for establishing this relationship are the same as those described in the ADD RECORD sentence. If a file to be INCLUDED is already associated with this record synonym, a new relationship will not be established.
- **RECORD NAME SYNONYM FOR FILE SYNONYM** - The EXCLUDE option dissociates this record and the designated file synonym. *File-name* must be a file synonym which is associated with this record synonym. For the INCLUDE option, rules for establishing this relationship are the same as those described in the ADD RECORD sentence. If a file synonym to be INCLUDED is already associated with this record synonym, a new relationship will not be established.
- **To alter a PREFIX/SUFFIX option** which has been designated with a record-to-file relationship, first EXCLUDE the record synonym to which the prefix or suffix is assigned. Then INCLUDE the record-to-file relationship with the new prefix or suffix. Record elements must be resubmitted in order to reflect the new element names.
- **RECORD NAME SYNONYM** - The EXCLUDE option deletes a synonym for this record. For the INCLUDE option, rules for formation of synonym names are the same as those for record primary names. The inclusion of record synonym names automatically builds record element synonym structures using the primary record element names with any prefix/suffix information appended.
- **OCCURRENCES** - The previously specified record occurrence count, if any, is replaced by *integer*. *Integer* may be 1 to 16 digits.
- **COMMENTS** - Rules for modifying comments are explained in the earlier discussion of the **Coding Format**. The existing comments, if any, are replaced by *comment* or deleted by NULL.

- **RECORD ELEMENT** - This clause associates an element with the preceding **RECORD** sentence. *Element-name* must be the primary name for a known element, or it must identify the element as a filler. The relationship between this element name and all record names (primary and synonyms) is established at this time. If the prefixed or suffixed name to be associated with the record is the primary element name plus the prefix or suffix, the concatenated element name is established automatically by the system at this time, along with the relationship.

**NOTE:** To define a filler as a record element, specify an element name of 'FIL $_{nnnn}$ ', where  $nnnn$  is the number of characters of filler. For example, to generate a filler of FILLER PIC X(7), code the following record element: 'FIL 0007'.

- **ELEMENT NAME SYNONYM** - This clause associates an element synonym and the primary record name. If *element-name* is not a known synonym for this element, it is established at this time.
- **ELEMENT NAME SYNONYM FOR RECORD SYNONYM...IS** - This option associates an element synonym and a record synonym. If the *element-name* is not a known synonym for this element, it is established at this time. If the prefixed or suffixed name to be associated with this record synonym is this element synonym plus the prefix or suffix, the concatenated element name is established at this time, along with the relationship. The concatenation of *record-name* and *version-no* must be a known synonym for this record.
- **PICTURE** - This clause is used only for group elements. When it is not present, subordinate elements will be automatically included in the record at this time. When it is present, the subordinate elements will not be included. Instead, an alphanumeric display picture will be included for the group element. Its length will be calculated automatically.
- **PICTURE IS** - This clause may be used for either group elements or primary (lowest level) elements. For group elements, the subordinate elements will not be included. Instead, the specified *character-string* will be included as the picture for the group element. For primary elements, the specified *character-string* will replace any previously specified picture.
- **USAGE** - This clause (re)designates the data usage for the element. When usage is **CONDITION-NAME**, a level number of 88 will be generated for this element. The default is **DISPLAY**. **USAGE** options follow.



# RECORD/RECORD ELEMENT

# RECORD/RECORD ELEMENT

USAGE option	Meaning	Acceptable to this processor
BIT	Bit string definition	IDMSDMLP
POINTER	Fullword Address Constant	IDMSDMLP
CONDITION-NAME	Level 88 values	IDMSDMLC
COMP (BINARY)	Binary	IDMSDMLC and IDMSDMLP
COMP-1 (SHORT POINT)	Short precision floating point	IDMSDMLC and IDMSDMLP
COMP-2 (LONG-POINT)	Long precision floating point	IDMSDMLC and IDMSDMLP
COMP-3 (PACKED)	Packed decimal	IDMSDMLC and IDMSDMLP
DISPLAY	Zoned decimal	IDMSDMLC and IDMSDMLP

- **REDEFINES** - This clause designates an element to redefine the preceding record element. *Element-name* must be the primary name for a known element.
- **OCCURS** - When the record element occurs a fixed number of times within this record, the OCCURS clause must specify the number of times it occurs. *Integer* may be one to four digits valued from 1 through 9999.
- **OCCURS DEPENDING ON** - When the record element occurs a variable number of times within this record, this clause must specify the maximum number of times it may occur. *Integer* may be one to four digits valued from 1 through 9999. *Element-name* must be a primary element name.
- **INDEXED BY** - This clause may be specified only when a record element is the subject of an OCCURS or OCCURS...DEPENDING ON clause. *Index-name* is stored away in the Dictionary and is inserted into a program's data division code by the COBOL processor as part of the record COPY function.
- **INDEX KEY** - This clause specifies the ordering of a multiply occurring element which is subordinate to this element. *Element-name* must be a primary element name. ASCENDING or DESCENDING defines the sequence of the ordered elements.
- **SYNC** - This clause will cause boundary alignment for the record element when it is copied into a source program according to the following rules:

If element USAGE is...	RECORD ELEMENT will be aligned on...
POINTER	FULLWORD
COMP (BINARY)	HALFWORD if S9(4) or less, else FULLWORD
COMP-1 (SHORT-POINT)	FULLWORD
COMP-2 (LONG-POINT)	DOUBLEWORD

- **COMMENTS** - Rules for coding comments are explained in the earlier discussion of the Coding Format.



NOTE: The following clauses are *not* used to define subordinate elements for this RECORD ELEMENT. The subordinate element-to-group element relationship is established in the ELEMENT sentence for the group element. The following clauses may be used to specify further length characteristics for multiply occurring subordinate elements, to specify alignment for the subordinate element, and to document the usage of index keys for tables (arrays).

- SUBORDINATE ELEMENT - A subordinate element-to-group element relationship must have been established between this element and the group element named in the RECORD ELEMENT clause. *Element-name* must be the primary element name. Subordinate record elements must be named in separate SUBORDINATE ELEMENT clauses, and must be input to the same sequence as they were coded in the SUBORDINATE ELEMENT clause in the ELEMENT sentence for the group.

NOTE: The OCCURS *integer* specified in the following clauses overrides the OCCURS *integer* clauses, if any, previously specified in the ELEMENT sentence.

- PICTURE - This clause is used only for group elements. When it is not present, subordinate elements will be automatically included in the record at this time. When it is present, the subordinate elements will not be included. Instead, an alphanumeric display picture will be included for the group element. Its length will be calculated automatically.
- PICTURE IS - This clause may be used for either group elements or primary (lowest level) elements. For group elements, the subordinate elements will not be included. Instead, the specified *character-string* will be included as the picture for the group element. For primary elements, the specified *character-string* will replace any previously specified picture.
- USAGE - This clause (re)designates the data usage for the element. When usage is CONDITION-NAME, a level number of 88 will be generated for this element. The default is DISPLAY. USAGE options follow.

USAGE option	Meaning	Acceptable to this processor
BIT	Bit string definition	IDMSDMLP
POINTER	Fullword Address Constant	IDMSDMLP
CONDITION-NAME	Level 88 values	IDMSDMLC
COMP (BINARY)	Binary	IDMSDMLC and IDMSDMLP
COMP-1 (SHORT POINT)	Short precision floating point	IDMSDMLC and IDMSDMLP
COMP-2 (LONG-POINT)	Long precision floating point	IDMSDMLC and IDMSDMLP
COMP-3 (PACKED)	Packed decimal	IDMSDMLC and IDMSDMLP
DISPLAY	Zoned decimal	IDMSDMLC and IDMSDMLP

## RECORD/RECORD ELEMENT

## RECORD/RECORD ELEMENT

- OCCURS - When the subordinate element is repeated a fixed number of times within this group element, this clause specifies the number of times it may occur. *Integer* may be one to four digits valued from 1 through 9999.
- OCCURS DEPENDING ON - When the subordinate element is repeated a variable number of times within this group element, this clause specifies the maximum number of times it may occur along with the DEPENDING ON *element-name*. *Integer* may be one to four digits valued from 1 through 9999. *Element-name* must be a primary element name.
- INDEXED BY - This clause may be specified only when a subordinate element is the subject of an OCCURS or OCCURS...DEPENDING ON clause. *Index-name* is stored away in the Dictionary and is inserted into a program's data division code by the COBOL processor as part of the record COPY function.
- INDEX KEY - This clause specifies the ordering of a multiply occurring element which is subordinate to this element. *Element-name* must be a primary element name. ASCENDING or DESCENDING defines the sequence of the ordered elements.
- SYNC - This clause will cause boundary alignment for the subordinate element when it is copied into a source program according to the following rules:

If element USAGE is...      SUBORDINATE RECORD ELEMENT will be aligned on...

POINTER	FULLWORD
COMP (BINARY)	HALFWORD if S9(4) or less, else FULLWORD
COMP-1 (SHORT-POINT)	FULLWORD
COMP-2 (LONG-POINT)	DOUBLEWORD

- COMMENTS - Rules for coding comments are explained in the earlier discussion of the Coding Format.
- REBUILD RECORD ELEMENTS - This subsentence (submitted instead of the RECORD ELEMENT subsentence) permits both *implicit* and *explicit* modification of individual record elements. *Implicitly*, modifications made to the (freestanding) elements *via* MODIFY ELEMENT syntax will be effected in the corresponding record elements if the REBUILD RECORD ELEMENTS subsentence is submitted. Individual record elements may be *explicitly* modified by means of subsequent RECORD ELEMENT subsentences, meanwhile preserving the remaining previously built record elements intact. RECORD ELEMENT clauses must be submitted in the sequence in which the elements occur. New elements may be included at the end of the record.

- RECORD ELEMENT - This clause associates an element with the preceding RECORD sentence. *Element-name* must be the primary name for a known element, or it must identify the element as a filler. The relationship between this element name and all record names (primary and synonyms) is established at this time. If the prefixed or suffixed name to be associated with the record is the primary element name plus the prefix or suffix, the concatenated element name is established automatically by the system at this time, along with the relationship.

NOTE: To define a filler as a record element, specify an element name of 'FIL*nnnn*', where *nnnn* is the number of characters of filler. For example, to generate a filler of FILLER PIC X(7), code the following record element: 'FIL 0007'.

- ELEMENT NAME SYNONYM - This clause associates an element synonym and the primary record name. If *element-name* is not a known synonym for this element, it is established at this time.
- ELEMENT NAME SYNONYM FOR RECORD SYNONYM...IS - This option associates an element synonym and a record synonym. If the *element-name* is not a known synonym for this element, it is established at this time. If the prefixed or suffixed name to be associated with this record synonym is this element synonym plus the prefix or suffix, the concatenated element name is established at this time, along with the relationship. The concatenation of *record-name* and *version-no* must be a known synonym for this record.
- PICTURE - This clause is used only for group elements. When it is not present, subordinate elements will be automatically included in the record at this time. When it is present, the subordinate elements will not be included. Instead, an alphanumeric display picture will be included for the group element. Its length will be calculated automatically.
- PICTURE IS - This clause may be used for either group elements or primary (lowest level) elements. For group elements, the subordinate elements will not be included. Instead, the specified *character-string* will be included as the picture for the group element. For primary elements, the specified *character-string* will replace any previously specified picture.
- USAGE - This clause (re)designates the data usage for the element. When usage is CONDITION-NAME, a level number of 88 will be generated for this element. The default is DISPLAY. USAGE options follow.

# RECORD/RECORD ELEMENT

# RECORD/RECORD ELEMENT

USAGE option	Meaning	Acceptable to this processor
BIT	Bit string definition	IDMSDMLP
POINTER	Fullword Address Constant	IDMSDMLP
CONDITION-NAME	Level 88 values	IDMSDMLC
COMP (BINARY)	Binary	IDMSDMLC and IDMSDMLP
COMP-1 (SHORT POINT)	Short precision floating point	IDMSDMLC and IDMSDMLP
COMP-2 (LONG-POINT)	Long precision floating point	IDMSDMLC and IDMSDMLP
COMP-3 (PACKED)	Packed decimal	IDMSDMLC and IDMSDMLP
DISPLAY	Zoned decimal	IDMSDMLC and IDMSDMLP

- **REDEFINES** - This clause designates an element to redefine the preceding record element. *Element-name* must be the primary name for a known element.
- **OCCURS** - When the record element occurs a fixed number of times within this record, the OCCURS clause must specify the number of times it occurs. *Integer* may be one to four digits valued from 1 through 9999.
- **OCCURS DEPENDING ON** - When the record element occurs a variable number of times within this record, this clause must specify the maximum number of times it may occur. *Integer* may be one to four digits valued from 1 through 9999. *Element-name* must be a primary element name.
- **INDEXED BY** - This clause may be specified only when a record element is the subject of an OCCURS or OCCURS...DEPENDING ON clause. *Index-name* is stored away in the Dictionary and is inserted into a program's data division code by the COBOL processor as part of the record COPY function.
- **INDEX KEY** - This clause specifies the ordering of a multiply occurring element which is subordinate to this element. *Element-name* must be a primary element name. ASCENDING or DESCENDING defines the sequence of the ordered elements.
- **SYNC** - This clause will cause boundary alignment for the record element when it is copied into a source program according to the following rules:

If element USAGE is...	RECORD ELEMENT will be aligned on...
POINTER	FULLWORD
COMP (BINARY)	HALFWORD if S9(4) or less, else FULLWORD
COMP-1 (SHORT-POINT)	FULLWORD
COMP-2 (LONG-POINT)	DOUBLEWORD

- **COMMENTS** - Rules for coding comments are explained in the earlier discussion of the Coding Format.



**NOTE:** The following clauses are *not* used to define subordinate elements for this RECORD ELEMENT. The subordinate element-to-group element relationship is established in the ELEMENT sentence for the group element. The following clauses may be used to specify further length characteristics for multiply occurring subordinate elements, to specify alignment for the subordinate element, and to document the usage of index keys for tables (arrays).

- **SUBORDINATE ELEMENT** - A subordinate element-to-group element relationship must have been established between this element and the group element named in the RECORD ELEMENT clause. *Element-name* must be the primary element name. Subordinate record elements must be named in separate SUBORDINATE ELEMENT clauses, and must be input to the same sequence as they were coded in the SUBORDINATE ELEMENT clause in the ELEMENT sentence for the group.

**NOTE:** The OCCURS *integer* specified in the following clauses overrides the OCCURS *integer* clauses, if any, previously specified in the ELEMENT sentence.

- **PICTURE** - This clause is used only for group elements. When it is not present, subordinate elements will be automatically included in the record at this time. When it is present, the subordinate elements will not be included. Instead, an alphanumeric display picture will be included for the group element. Its length will be calculated automatically.
- **PICTURE IS** - This clause may be used for either group elements or primary (lowest level) elements. For group elements, the subordinate elements will not be included. Instead, the specified *character-string* will be included as the picture for the group element. For primary elements, the specified *character-string* will replace any previously specified picture.
- **USAGE** - This clause (re)designates the data usage for the element. When usage is CONDITION-NAME, a level number of 88 will be generated for this element. The default is DISPLAY. USAGE options follow.

USAGE option	Meaning	Acceptable to this processor
BIT	Bit string definition	IDMSDMLP
POINTER	Fullword Address Constant	IDMSDMLP
CONDITION-NAME	Level 88 values	IDMSDMLC
COMP (BINARY)	Binary	IDMSDMLC and IDMSDMLP
COMP-1 (SHORT POINT)	Short precision floating point	IDMSDMLC and IDMSDMLP
COMP-2 (LONG-POINT)	Long precision floating point	IDMSDMLC and IDMSDMLP
COMP-3 (PACKED)	Packed decimal	IDMSDMLC and IDMSDMLP
DISPLAY	Zoned decimal	IDMSDMLC and IDMSDMLP



- OCCURS - When the subordinate element is repeated a fixed number of times within this group element, this clause specifies the number of times it may occur. *Integer* may be one to four digits valued from 1 through 9999.
- OCCURS DEPENDING ON - When the subordinate element is repeated a variable number of times within this group element, this clause specifies the maximum number of times it may occur along with the DEPENDING ON *element-name*. *Integer* may be one to four digits valued from 1 through 9999. *Element-name* must be a primary element name.
- INDEXED BY - This clause may be specified only when a subordinate element is the subject of an OCCURS or OCCURS...DEPENDING ON clause. *Index-name* is stored away in the Dictionary and is inserted into a program's data division code by the COBOL processor as part of the record COPY function.
- INDEX KEY - This clause specifies the ordering of a multiply occurring element which is subordinate to this element. *Element-name* must be a primary element name. ASCENDING or DESCENDING defines the sequence of the ordered elements.
- SYNC - This clause will cause boundary alignment for the subordinate element when it is copied into a source program according to the following rules:

If element USAGE is...	SUBORDINATE RECORD ELEMENT will be aligned on...
POINTER	FULLWORD
COMP (BINARY)	HALFWORD if S9(4) or less, else FULLWORD
COMP-1 (SHORT-POINT)	FULLWORD
COMP-2 (LONG-POINT)	DOUBLEWORD

- COMMENTS - Rules for coding comments are explained in the earlier discussion of the Coding Format.
- REMOVE RECORD ELEMENT - The submission of the REMOVE RECORD ELEMENT subsentence subsequent to the REBUILD RECORD ELEMENTS subsentence permits the *explicit* removal of individual record elements. (The elements themselves, as freestanding dictionary entities, are *not* deleted thereby.)

**NOTE:** Schema record elements cannot be removed by means of the REMOVE RECORD ELEMENT subsentence.

## APPENDIX H

### SAMPLE DATABASE DEFINITION

This appendix presents a example of a complete database definition. IDD syntax listings of an actual application illustrate the results of Phases I, II, and III of the database development process.

Included in the appendix are:

1. Subsystem Specification
2. Service analysis
3. Element definitions
4. Element group definitions
5. Data entity definitions
6. Schema definition.

NAVAL INTELLIGENCE COMMAND

SECURITY MANAGEMENT SUBSYSTEM (81-52-01)

of the

NAVAL INTELLIGENCE COMMAND ON-LINE SYSTEM (NICOLS)

SUBSYSTEM SPECIFICATION

23 January 1979

Prepared for:  
Naval Ocean Systems Center  
271 Catalina Blvd.  
San Diego, California 92152

Prepared by:  
SDC Integrated Services, Inc.  
3065 Rosecrans Place, Suite 102  
San Diego, California 92110

Security Management

Subsystem Specification

## TABLE OF CONTENTS

SECTION	PAGE
1 GENERAL . . . . .	1
1.0 Introduction . . . . .	1
1.1 Purpose of Subsystem Specification . . . . .	1
1.2 Project References . . . . .	1
2 SUMMARY OF REQUIREMENTS . . . . .	3
2.0 Background . . . . .	3
2.1 Subsystem Description . . . . .	3
2.1.1 General . . . . .	3
2.1.2 Sub-Subsystem Description . . . . .	4
2.1.2.1 Security Access Control . . . . .	4
2.1.2.2 Badge Control . . . . .	5
2.1.2.3 Facility Accreditation . . . . .	5
2.1.2.4 Visitor Control . . . . .	5
2.2 Subsystem Functions . . . . .	5
2.2.1 Information Management in the Batch Mode . . . . .	6
2.2.2 Information Management in the Online Mode . . . . .	15
2.2.3 Accuracy and Validity . . . . .	18
2.2.4 Timing . . . . .	18
2.3 Flexibility . . . . .	18
3 ENVIRONMENT . . . . .	20
3.1 Equipment Environment . . . . .	20
3.2 Support Software Environment . . . . .	20
3.3 Interfaces . . . . .	20
3.4 Security . . . . .	21
3.5 Controls . . . . .	21

3-76

## Security Management

## Subsystem Specification

### SECTION 1. GENERAL.

1.0 Introduction. This document presents detailed design information on the Security Management Subsystem(s).

1.1 Purpose of Security Management Subsystem (SECSS) Specification.

This document specifies the design for the Security Management Subsystem. It is written to fulfill the following objectives:

- a. To provide a detailed definition of the subsystem functions.
- b. To communicate details of the on-going analysis to the user's operational personnel.
- c. To define, in detail, the interfaces with other systems and subsystems and the facilities to be utilized for accomplishing the interface.

### 1.2 Project References

1. Cullinane Corporation, "CDMS--A Visual Presentation", Cullinane Data Management System.
2. Cullinane Corporation, "Culprit--Information Retrieval System".
3. Cullinane Corporation, "Data Base Design and Definition Guide Release 4.5", October 1977.
4. Cullinane Corporation, "Data Base Design and Definition Guide Release 5.0", August 1978.
5. Cullinane Corporation, "IDD--Integrated Data Dictionary".
6. Cullinane Corporation, "IDMS Concepts and Facilities", 1977.
7. Cullinane Corporation, "IDMS Culprit--User's Guide"
8. Cullinane Corporation, "IDMS Installation and Operations Guide Release 4.5", September 1977.
9. Cullinane Corporation, "IDMS On Line Query-OLQ", November 1977.



## Security Management

## Subsystem Specification

10. Cullinane Corporation, "IDMS Programmer's Reference Guide Release 4.5", Part I and II, November 1977.
11. Cullinane Corporation, "IDMS Sequential Processing Facility Release 4.5", April 1978.
12. Cullinane Corporation, "IDMS--Error Codes and Messages Guide", January 1978.
13. Cullinane Corporation, "IDMS--Integrated Data Dictionary Users Guide Release 1.0", August 1977.
14. Cullinane Corporation, "IDMS-DC Presentation", October 1978.
15. Cullinane Corporation, "Shadow II Cobol Programmer's Reference Manual, Version 2", December 1977.
16. Cullinane Corporation, "Shadow II Operations Manual, Version 2 OS", January 1978.
17. Cullinane Corporation, "Shadow II Programmer's Guide-DOS, Version 2", December 1977.
18. Cullinane Corporation, "Shadow II System Generation and Maintenance Manual, Version 2", February 1978.
19. Cullinane Corporation, "The Cullinane Data Management System", 1978.
20. Cullinane Corporation, "User's Guide Release 1.2", November 1977.
21. Defense Intelligence Agency, "Authorized Data Elements and Related Features", Volume I (A-M) and Volume II (N-Z), July 1976.
22. NAVINTCOM Document Number 39N001 FD-01B, "Integrated Automated Intelligence Processing System Specification", Functional Description, April 1978 (SECRET).
23. NAVINTCOM INST 5200.3A, "Data Base Development and Design Guide", 7 August 1978 (Version 2 dated November 1978).

## SECTION 2. SUMMARY OF REQUIREMENTS.

## 2.0 Background.

The Special Security/Special Activities Division (NIC-44) of the Naval Intelligence Command Headquarters is tasked with the maintenance and accounting of security clearances and persons possessing security clearances. The control of access to NAVINTCOM locations, the control of badges, and the accreditation of secure areas must also be maintained.

Batch processing facilities have supported these requirements for several years. Increasing requirements for administering and controlling security access has outstripped the ability of current ADP software to perform the required functions. On-line, interactive access to the Security Management database is required, making all security-related information immediately available.

## 2.1 Subsystem Description.

A complete description of the Security Management Subsystem is presented in the following paragraphs.

## 2.1.1 General.

The Security Management Subsystem consists of equipment and facilities located at Suitland, Maryland, with communication links to remote terminal locations scattered in the Washington D.C. area.

The functions of the Security Management Subsystem are divided into four major areas. They are:

## a. Security Access Control.

The Security Management Subsystem shall provide for the maintenance and the accounting for all desired security classifications (Access codes), the billets associated with each command, the incumbents of the billets, the access history of incumbents, and the allocation of restricted access codes.

## b. Badge Control.

The Security Management Subsystem shall provide for the maintenance and the accounting for an

## Security Management

## Subsystem Specification

unlimited number or type of badges for use by personnel associated with the various Naval Intelligence activities.

The Badge Control function will be designed in to the system under the current tasking, but will not be implemented as part of this effort.

c. Facility Accreditation.

The Security Management Subsystem shall provide the capability to maintain accounting of the accreditation (security levels), with technical surveillance counter measure results and inspection dates and remarks, of all facilities under control of the Naval Intelligence Command.

d. Visitor Control.

The Security Management Subsystem shall provide the maintenance capability for the accounting of visitors to Naval Intelligence facilities at an unlimited number of locations.

The Visitor Control function will be designed in to the system under the current tasking, but will not be implemented as part of this effort.

2.1.2 Subsystem Area Descriptions.

The Security Management Subsystem is composed of the following areas:

Security Access Control

Badge Control

Facility Accreditation

Visitor Control

These areas are described in the following paragraphs.

2.1.2.1 Security Access Control.

The Security Access Control portion of the subsystem concerns itself with billets and incumbents and their associated SI access levels.

The main sources of input data (adds, deletes, indoctrinations, and debriefings) are message tapes from all of the Navy activities. Erroneous data from the message tapes shall be on a report in a form that will be suitable for final message preparation by the user.

As people are indoctrinated into various accesses, a history file shall be maintained to provide future accessibility of the information.

Individual controls shall be provided to maintain restrictions on the total number of access tickets of each type in use.

Online and batch capability for inquiry and updating of all the Security Access information will be provided.

#### 2.1.2.2 Badge Control.

The Badge Control Phase of the Security Management Subsystem is concerned with the issuing and control of all badges by the various activities within Naval Intelligence Command.

Inquiry and updating functions for this information shall be implemented in a subsequent task.

#### 2.1.2.3 Facility Accreditation.

The Facility Accreditation phase of the Security Management Subsystem is concerned with the inspection of all physical spaces that are to be accredited.

Online and batch capability for inquiry and updating of all the Facility Accreditation information will be provided.

#### 2.1.2.4 Visitor Control.

The Visitor phase of the Security Management Subsystem is concerned with the control and history of visitors into all facilities within Naval Intelligence Command.

Inquiry and updating functions for this information shall be implemented in a subsequent task.

## 2.2 Subsystem Functions

The Security Management Subsystem shall provide maintenance (updates) to include the addition, change, deletion and relationships of records in the database. The subsystem shall automatically record the date that a record in the database is added or changed. The subsystem shall also provide printed reports from the database in both batch and on-line modes. Specific reports and functions available in the subsystem are identified below.

### 2.2.1 Information Management in the Batch Mode.

The following printed reports, 132 characters per line, shall be available upon request of the Security Management Subsystem authorized user. Reporting at Remote Job Entry (RJE) stations will be limited to single copy listings on 8 1/2 by 14 inch paper (maximum length of 5000 lines, at 8 lines per inch). Reports requiring other forms, reports of extended length, or reports at 6 lines per inch, will be printed at the NAVINTCOM ADP center in Suitland and delivered by courier to NIC-44 (unless specified to be delivered elsewhere).

#### 2.2.1.1 Update of Billet Related Data.

The billet related data in the database may be updated to include additions, changes and deletions, and to also define relationships to incumbents. A printed transaction report showing all transactions made in the data base will be produced. In addition, a tickler listing of those billets that have incumbents in the process of being 'read out' is produced.

#### 2.2.1.2 Update of Incumbent Related Data.

The incumbent related data in the database may be updated to include additions, changes and deletions, and to also define relationships to billets. A printed transaction report showing all transactions in the database shall be produced.

#### 2.2.1.3 Command List Reports.

The Master Command Alpha List Report contains all incumbents in the database, listed in alphabetical order by name.



In addition, variations of this report may also be produced (on demand).

a. The report can be produced for only those incumbents with NIPSSA billets. NIPSSA Billets are defined by a series of command code/billet number ranges. Those ranges are:

223000/000 through 223000/999,  
280232/000 through 280232/999, and  
280290/000 through 280290/999.

The report is sorted in alphabetic order by name. NIPSSA-01S is the recipient of this report.

b. An alternate NIPSSA report (using the same ranges as item a) is sorted in background investigation date order. NIPSSA-01S is the recipient of this report.

c. A Master Command Alpha List - Series 800XXX (previous job id #34). Selects for the 800000-800999 command code series and lists in alphabetic order by name.

d. A Master Command List - Series 800XXX (previous job id #35). Selects the 800000-800999 command code series listed in command code/billet number order.

e. A Master Command List Report (previous job id #39). Selects a range of command codes as specified by the user (which may be the total file) and lists them in command code/billet number sequence with page breaks when the command code changes.

#### 2.2.1.4 NAVSIM Command Alpha List Report (previous job id #32).

The NAVSIM Command Alpha List Report contains all incumbents in the database who possess AN, AX, ES, EX, CR, C, CY, CL, MA, DN, PW, PC, or PM accesses, listed in alphabetic order by name.

#### 2.2.1.5 NAVSIM SSO Admin Report (previous job id #40).

The NAVSIM SSO Admin Report contains all incumbents in command codes 010000 through 100000 or 200000

through 999999 who possess AN, AX, ES, EX, CR, CX, CY, CL, MA, DN, PW, PC, or PM accesses, listed in command code/billet number order.

2.2.1.6 SI Command Alpha List Report (previous job id #33).

The SI Command Alpha List Report contains all incumbents in the database who possess SI accesses, listed in alphabetic order by name.

2.2.1.7 Old BI Command Alpha List Report (previous job id #36).

The Old BI Command Alpha List Report contains all incumbents with Navy command codes (200000-299999 or 700000-799999) in the database who possess SI accesses more than 4 1/2 years old, listed in alphabetic order by name. An alternate report with the same information is:

Old BI Command List Report (previous job id #42).  
Listed in command code/ billet number sequence.

2.2.1.8 Master Command Billet List Report (previous job id #43).

The Master Command Billet List contains all billets in all commands in the database, listed in command code/billet number order.

2.2.1.9 Master Command Titles/Billet Report (previous job id #46).

The Master Command Titles/Billet Report is for all commands on file, listed in command code/billet number order. An alternate sort is available in the,

Master Alpha Command Titles/Billet Report (previous job id #49). Listed in alphabetic order by name.

2.2.1.10 Master Command Incumbent Totals Reports (previous job id #47).

The Master Command Incumbent Totals Reports are available in four options. All of the options are sorted in command code sequence. Reports defined by items a, b, and c exclude all Z accesses.

## Security Management

## Subsystem Specification

a. Master Command Totals - Incumbent Navy Only. This report contains totals of all accesses held by incumbents, subtotaled by command code, and grand totaled for all Navy commands. Assigned command codes for the Navy are in the following ranges:

200000 through 299999,  
 722000 through 722999,  
 732000 through 732999,  
 742000 through 742999,  
 772000 through 772999,  
 774000 through 774999, and  
 782000 through 782999.

b. Master Command Totals - Incumbent Non-Navy Only. This report contains totals of all accesses held by incumbents, subtotaled by command code, and grand totaled for all non-Navy commands. These command codes are all those not defined in the ranges for item a, with the exception that command codes 800000 through 899999 are excluded.

c. Master Command Totals - 800000 Series. This report contains totals of all accesses held by incumbents, subtotaled by command code, and grand totaled for all command codes in the range 800000 through 899999.

d. Master Command Totals - Incumbent Z Data. This report contains totals of all Z accesses held by incumbents, subtotaled by command code, and grand totaled for all command codes.

#### 2.2.1.11 Master Command Billet Totals Reports (previous job id #48).

The Master Command Billet Totals Reports are available in four options. All of the options are sorted in command code sequence. Reports defined by items a, b, and c exclude all Z accesses.

a. Master Command Totals - Billet Navy Only. This report contains totals of all accesses assigned to billets, subtotaled by command code, and grand totaled for all Navy commands. Assigned command codes for the Navy are in the following ranges:

## Security Management

## Subsystem Specification

200000 through 299999,  
 722000 through 722999,  
 732000 through 732999,  
 742000 through 742999,  
 772000 through 772999,  
 774000 through 774999, and  
 782000 through 782999.

b. Master Command Totals - Billet Non-Navy Only. This report contains totals of all accesses assigned to billets, subtotaled by command code, and grand totaled for all non-Navy commands. These command codes are all those not defined in the ranges for item a, with the exception that command codes 800000 through 899999 are excluded.

c. Master Command Totals - 800000 Series. This report contains totals of all accesses assigned to billets, subtotaled by command code, and grand totaled for all command codes in the range 800000 through 899999.

d. Master Command Totals - Billet Z Data. This report contains totals of all Z accesses assigned to billets, subtotaled by command code, and grand totaled for all command codes.

#### 2.2.1.12 Master Social Security List Report (previous job id #55).

The Master Social Security List Report contains all incumbents in the database, listed in social security number order.

#### 2.2.1.13 Z Command Alpha List Report (previous job id #56).

The Z Command Alpha List Report contains all incumbents in the database who possess Z or MED accesses listed in alphabetic order by name. Options of this report are,

a. Z Command Alpha List Report - MED Accesses Only (previous job id #62). This report is produced containing only incumbents with MED accesses.

b. Z Command Alpha List Report - DUL Accesses Only (previous job id #53). This report is the same as

## Security Management

## Subsystem Specification

item a, except that only DUL access are used for selection.

#### 2.2.1.14 Z Command List Report (previous job id #57).

The Z Command List Report contains all incumbents in the database who possess Z or MED accesses listed in command code/billet number order. There is a page break when the command code changes. An option for this report is,

Z Command List Report - MED Accesses Only (previous job id #61). This report contains information on incumbents with MED accesses. The sort is alphabetically by name within command code.

#### 2.2.1.15 Incumbent B Access Report.

The Incumbent B Access Report contains all incumbents in the database who possess any type of B access, listed in alphabetical sequence by name.

#### 2.2.1.16 Billet B Access Report.

The Billet B Access Report contains all billets in the database which are assigned any type of B access, listed in command code/billet number sequence.

#### 2.2.1.17 Billet/Incumbent Merge Report.

The Billet/Incumbent Merge Report contains all billet and incumbent data for the commands defined by the user supplied selection parameter, listed in command code/billet number sequence.

#### 2.2.1.18 Clearance Deletion Report (Incumbent).

All Data base records which contain no accesses after indicated accesses are deleted are listed on this report. The Clearance Deletion Report also gives record counts.

#### 2.2.1.19 DIA Punched Cards.

DIA punched cards are punched onto the DIA Punched Card Tape to be punched by NIPSSA. DIA punched cards are produced at the request of the user. One punched card is produce for each command that begins



with a '2' and has an SAO access code of 'A'. The tape contains one record for each billet in the database and for each incumbent in the incumbent database whose command begins with a '2' and who possess a TK, BB, BI, BE, BG, BR, BH, or PO access. This job is normally run Quarterly.

#### 2.2.1.20 Administrative Reports.

Administrative Reports are available in four options. All of the options are sorted in SSO command code, command code, and billet number sequence.

a. SSO Administrative Report (previous job id #52). The SSO Administrative Report contains all billets with their associated incumbents and accesses (excluding Z accesses) for the command codes specified below. Caution: This report is extremely long and should be scheduled with the ADP Center.

010000 through 199999,  
219800 through 269999,  
270500 through 789999, and  
991000 through 999000.

One option of this report is produced using only the billets selected by specific command code ranges that are entered by the user.

A second option of this report is produced using only the billets assigned Z accesses.

Another option of the report defined in item a above, is of the NIPSSA billets. NIPSSA Billets are defined by a series of command code/billet number ranges. Those ranges are:

223000/000 through 223000/999,  
280232/000 through 280232/999, and  
280290/000 through 280290/999.

This report is delivered to NIPSSA-01S.

b. Access Management Report. This report is a listing, of billets with incumbents, that contains only those billets that have been indicated as temporary (for 'floating' access management). The billets to be selected must also be within the

command code range supplied by the user. The report is sorted in command code and billet number sequence.

c. SSO Administrative Report - 800XXX Series (previous job id #37). This report contains only 800000 through 800999 command codes with their associated billets and incumbents, listed in command code and billet number sequence.

d. SSO Administrative Report - DUL Accesses Only (previous job id #54). This report contains all billets and incumbents in the database who possess DUL accesses, listed in command code/billet number sequence.

e. OPNAV Administrative Report (previous job id #58). The OPNAV Administrative Report contains all billets with their associated incumbents and accesses (excluding Z accesses) for the command codes specified below, listed in SSO command, command code, and billet number order. Command codes used for this report are:

742000-742999,  
772000-772999,  
774000-774999,  
782000-782999, and  
999900-999999.

#### 2.2.1.21 Update of Access History Related Data.

The access history data in the database may be updated to include additions, change and deletions. A printed transaction report showing all transactions made to the data base will be produced.

#### 2.2.1.22 Retrieval of Access History Related Data.

This report contains all access history data that is maintained in the database for the specific name as supplied by the user.

#### 2.2.1.23 Dynamic Utilization (Float) of Access Codes.

The ability to dynamically allocate the access codes ('tickets') among all active billets will be provided automatically for a specific billet when the billet is added to the file. This control will

## Security Management

## Subsystem Specification

provide for optimum usage of the allowed number of 'tickets' for which the Navy has responsibility.

#### 2.2.1.24 Access Management Report.

The Access Management Report is a listing of billets with incumbents that contains only those billets that have been indicated as temporary (for 'floating' access management). The billets to be selected must also be within the command code range supplied by the user. The report is sorted in command code and billet number sequence.

#### 2.2.1.25 Update of Facilities Accreditation Data.

The Facilities Accreditation data in the database may be updated to include additions, change and deletions. A printed transaction report showing all transactions made to the database will be produced.

#### 2.2.1.26 List of Spaces Without Final Accreditation.

This report will produce a list of all records in which the final accreditation is blank. The report is used by NIC-44 to determine why spaces have not received their final accreditation.

#### 2.2.1.27 Facilities Accreditation List of Spaces.

This report will produce a list of all rooms that are accredited. The report is in command code/building number/room number sequence.

#### 2.2.1.28 Facilities Accreditation Reinspection Report.

This report lists all spaces that are due for reinspection in the next three months. A date, entered by the user, which specifies the starting date is entered into the system. The report is in command code/building number/room number sequence.

#### 2.2.1.29 Facilities Accreditation Semi-Annual Report.

This report is a list of all spaces to be used by parent activities to keep a record of their spaces. The report is in command code/building number/room number sequence with page breaks at command code changes. The commands to be printed are entered by the user as a low to high command code number range.

23 January 1979

Page 1

## Security Management

## Subsystem Specification

### 2.2.1.30 Facility List Report.

This report produces a list of all spaces having '00' in the last two characters of the control number. This report is ordered by control number and is produced upon request of the user.

### 2.2.1.31 Facilities Accreditation TSCM Required Spaces.

This report produces a list of all spaces requiring Technical Surveillance Counter Measures (TSCM) and is produced upon user request.

### 2.2.2 Information Management in the Online Mode.

The following online access capabilities will be made available by the Security Management Subsystem to the authorized user. Data will be accessible, for either updating or inquiry, by means of computer terminals connected to the main computer. Hard copy of the screen images, 24 lines of 80 characters, will be available at the users location. The online reports that are generated (batch capture) can be a maximum of 10 pages for each request. Each page consists of 20 lines with 80 characters per line. If the user has a request that generates more than the 10 page limit, only the first 10 pages will be made available. A subsequent request, that specifies the beginning and ending sequences for the data block in interest, will have to be entered (by the user) to view more information.

#### 2.2.2.1 Update of Billet Related Data.

The billet related data in the database may be updated to include additions, changes and deletions, and to also define relationships with Incumbents. A hard copy of the transaction may be produced at the option of the user. The input of data can be several sequentially accessed formatted screens. The screen is defined to be 24 lines of 80 characters on the display terminal.

#### 2.2.2.2 Update of Incumbent Related Data.

The incumbent related data in the database may be updated to include additions, changes and deletions, and to also define relationships with Billets. A

23 January 1979

Page 15

## Security Management

## Subsystem Specification

hard copy of the transaction may be produced at the option of the user. The input of data can be several sequentially accessed formatted screens.

#### 2.2.2.3 Retrieval of User Defined Data.

Any data in the Security Management Subsystem database may be displayed to the user by a user specified query. This query defines to the system the specific data that the user requires. A hard copy of the information may be produced at the option of the user. The output of data can be displayed on several sequentially accessed formatted screens.

#### 2.2.2.4 Selective Billet Report.

This report contains all billets and incumbents in the database within the user defined from-and-through parameter. The from-and-through parameter for this request consists of the low and the high command-billet numbers that determine the range to be selected. It is sorted in command code and billet number sequence. Data for this report includes both incumbent, billet, and access data.

#### 2.2.2.5 Selective Billet Report by Incumbent.

This report contains all billets and incumbents in the database within the user defined from-and-through parameter. The from-and-through parameter for this request consists of the low and the high incumbent names that determine the alphabetic range to be selected. It is sorted alphabetically by incumbent name. Data for this report includes both incumbent, billet, and access data.

#### 2.2.2.6 Selective Billet Report by Social Security Number.

This report contains all billets and incumbents in the database within the user defined from-and-through parameter. The from-and-through parameter for this request consists of the low and the high social security numbers that determine the range to be selected. It is sorted numerically by the social security number. Data for this report includes both incumbent, billet, and access data.



23 January 1979

Page 3

Security Management

Subsystem Specification

2.2.2.7 Access Code Summary Report.

This report contains a summary of the number of accesses in use for all types of accesses. It is listed alphabetically by access code. Data for this report uses billet and access code data.

2.2.2.8 Floating Access Code Summary Report.

This report contains a detailed status of floating control information for all types of accesses. It is listed alphabetically by access code. Data for this report uses billet and access code data.

2.2.2.9 Branch of Service Summary Report.

This report contains a summary of the number of accesses in use for all types of access by branch of service in the database. It is listed alphabetically by access code. Data for this report uses incumbent and access data.

2.2.2.10 Update of Access History Related Data.

The access history related data in the database may be updated to include additions, changes and deletions. A hard copy of the transaction may be produced at the option of the user. The input of data can be several sequentially accessed formatted screens.

2.2.2.11 Selective Access History Report by Incumbent.

This report contains all access history data that is maintained in the database for the specific name as supplied by the user. The requester can view the report at the users terminal after the query is entered.

2.2.2.12 Dynamic Utilization (Float) of Access Codes.

The ability to dynamically allocate the access codes ('tickets') among all active billets will be provided automatically for a specific billet when the billet is added to the file. This control will provide for optimum usage of the allowed number of 'tickets' for which the Navy has responsibility.

2.2.2.13 Update of Facilities Accreditation Data.

23 January 1979

Page 17

23 January 1979

Page 4

## Security Management

## Subsystem Specification

The facilities accreditation related data in the database may be updated to include additions, changes and deletions. A hard copy of the transaction may be produced at the option of the user. The input of data can be several sequentially accessed formatted screens.

### 2.2.3 Accuracy and Validity

Transaction validation shall be performed for all incoming data. Inconsistent or invalid transactions will be rejected and update reports returned to the user for resolution.

### 2.2.4 Timing

Response time is defined as the time lapse between a user generated demand or request and the subsystem output to the demand or request. The response time will vary according to the priority of the requested action and the length of the procedures required to accomplish the requested action.

Incoming requests shall be processed on a priority basis (highest priority requests first), with requests of equal priority processed in a first-in, first-out order. The response time for these requests will be dependent on the response time of the Operating System and the efficiency (size and execution time) of the applications software code mix. The response time for lower priority requests will be strongly dependent on the number of higher requests which will be processed first.

Average acknowledgement time for a typical online query will be approximately 10 seconds. The actual response time is a function of the complexity of the query. Generation of a summary report in response to an online user request will average from 15 to 120 minutes (using batch capture). The time required to respond to a batch query will typically range from 2 to 48 hours, depending on the priority assigned to the query by the user.

## 2.3 Flexibility

The subsystem design reflects previously established requirements with the addition of interactive terminals and some minor formatting changes. Due to the

23 January 1979

Page 18

evolutionary nature of users' requirements, the software shall be designed for ease of modification with minimal impact on system use. This will facilitate the future implementation of enhanced subsystem capabilities, such as the Badge and Visitor Control phases described in Sections 2.1.2.2 and 2.1.2.4 above.

Another expansion to the subsystem could be a more comprehensive message tape processor that would be capable of extracting valid information from messages containing only minor errors in format.

Billet and Incumbent history could be maintained offline on magnetic tape or on printed documents.

The Subsystem will have two main interfaces with the user. They are:

a. Batch Processing. All data will be accessible for updating at the main computer facility or by means of Remote Job Entry (RJE) stations. Reporting at RJE stations will be limited to single copy listings on 8 1/2 by 14 inch paper (maximum length of 5000 lines, at 8 lines per inch). Reports requiring other forms, multiple copies, printing at 6 lines per inch, or reports of extended length, will be printed at the Suitland facility and delivered by courier.

b. Online Processing. Data will be accessible, for either updating or inquiry, by means of computer terminals connected to the main computer through encrypting devices. Hard copy of the screen images, 24 lines of 80 characters, will be available at the users location.

### 3.4 Security

Overall the subsystem will operate at the SECRET level. However, each data element will have its own clearance level. This will provide query capability at the unclassified level where appropriate.

### 3.5 Controls

Access to the Security Management Subsystem within the NICOLS database will be under the control of the Head of NIC-44. Upon direction, NIPSSA-03DN will update the Responsibility clauses in the Integrated Data Dictionary that grant update, modify, delete, and inquiry permissions for the elements and records of the Subsystem.

## SECTION 3. ENVIRONMENT.

## 3.1 Equipment Environment

The ADPE Systems currently in use within the NAVINTCOM ADP Center in Suitland are:

- a. IBM 360/50 computer, 1 megabyte memory, 7- and 9-track tape units, card reader-punch, 1100 lpm printer with replaceable print train.
- b. COMTEN 3650 terminal controller, supporting multiple medium-speed binary synchronous communications lines.
- c. CDC 27801 remote-job-entry terminal, primarily for support of software development efforts. This terminal will be shared with NAVINTCOM programming personnel working on other ADP-related tasks.
- d. IBM 3277 (compatible) terminals and 3284 (compatible) printers, primarily for user access to the subsystem.

## 3.2 Support Software Environment

The principle software currently in use within the NAVINTCOM ADP center environment are:

- a. OS/MVT/V21.8F operating systems with HASP spooling support.
- b. Integrated Database Management System (IDMS) database management system with COBOL preprocessor. Version 4.5 or higher.
- c. Integrated Data Dictionary (IDD) dictionary package supporting IDMS.
- d. SHADOW II teleprocessing monitor and associated COBOL preprocessor.
- e. CULPRIT report writer and query support package, version 4.5 or higher.
- f. IBM ANSII COBOL language compiler, Version 4.

## 3.3 Interfaces



MEMBER NAME DIASMSUP  
 ADD PROGRAM NAME IS S8152001  
 PREPARED BY 'ISI'  
 PROGRAM DESCRIPTION  
 'UPDATE OF BILLET RELATED DATA'  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-UNCLASSIFIED  
 OUTPUT-SECURITY IS OUT-CONFIDENTIAL  
 WITHIN SUBSYSTEM SECSS  
 OUTPUT-MODE IS REPORT-MODE  
 FREQUENCY IS DAILY  
 SERVICE-PRIORITY IS NORMAL-PRIORITY  
 RESPONSE-TIME IS OVERNIGHT-RESPONSE  
 MODE IS BATCH  
 POINT-OF-CONTACT IS 'CHIEF ROWE'  
 POC-PHONE IS '325-0880'  
 USER-ORGANIZATION IS 'NIC-44'  
 SERVICE-HISTORY IS EXISTING-SERVICE  
 COMMENTS  
 - 'THE BILLET RELATED DATA IN THE DATABASE WILL  
 - 'BE UPDATED TO INCLUDE ADDITIONS, CHANGES AND  
 - 'DELETIONS, AND TO ALSO DEFINE RELATIONSHIPS  
 - 'TO INCUMBENTS. A PRINTED TRANSACTION REPORT  
 - 'SHOWING ALL TRANSACTIONS MADE IN THE DATA BASE  
 - 'WILL BE PRODUCED. IN ADDITION, A TICKLER  
 - 'LISTING OF THOSE BILLETS THAT HAVE INCUMBENTS  
 - 'IN THE PROCESS OF BEING READ OUT IS PRODUCED..'  
 ADD PROGRAM NAME IS S8152002  
 PREPARED BY 'ISI'  
 PROGRAM DESCRIPTION  
 'UPDATE OF INCUMBENT RELATED DATA'  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-UNCLASSIFIED  
 OUTPUT-SECURITY IS OUT-CONFIDENTIAL  
 WITHIN SUBSYSTEM SECSS  
 OUTPUT-MODE IS REPORT-MODE  
 FREQUENCY IS DAILY  
 SERVICE-PRIORITY IS NORMAL-PRIORITY  
 RESPONSE-TIME IS OVERNIGHT-RESPONSE  
 MODE IS BATCH  
 POINT-OF-CONTACT IS 'CHIEF ROWE'  
 POC-PHONE IS '325-0880'  
 USER-ORGANIZATION IS 'NIC-44'  
 SERVICE-HISTORY IS EXISTING-SERVICE  
 COMMENTS  
 - 'THE INCUMBENT RELATED DATA IN THE DATABASE  
 - 'WILL BE UPDATED TO INCLUDE ADDITIONS, CHANGES  
 - 'AND DELETIONS, AND TO ALSO DEFINE RELATIONSHIPS  
 - 'TO BILLETS. A PRINTED TRANSACTION REPORT  
 - 'SHOWING ALL TRANSACTIONS IN THE DATABASE SHALL  
 - 'BE PRODUCED..'  
 ADD PROGRAM NAME IS S8152003  
 PREPARED BY 'ISI'  
 PROGRAM DESCRIPTION  
 'MASTER COMMAND ALPHA LIST REPORT'  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-UNCLASSIFIED  
 OUTPUT-SECURITY IS OUT-CONFIDENTIAL  
 WITHIN SUBSYSTEM SECSS

MEMBER NAME DIASMSUB  
 OUTPUT-MODE IS REPORT-MODE  
 FREQUENCY IS ADHOC  
 SERVICE-PRIORITY IS NORMAL-PRIORITY  
 RESPONSE-TIME IS OVERNIGHT-RESPONSE  
 MODE IS BATCH  
 POINT-OF-CONTACT IS 'CHIEF ROWE'  
 POC-PHONE IS '325-0880'  
 USER-ORGANIZATION IS 'NIC-44'  
 SERVICE-HISTORY IS EXISTING-SERVICE  
 COMMENTS  
 'THE MASTER COMMAND ALPHA LIST REPORT  
 - 'CONTAINS ALL INCUMBENTS IN THE DATABASE.  
 - 'LISTED IN ALPHABETICAL ORDER BY NAME.''  
 ADD PROGRAM NAME IS S8152004  
 PREPARED BY 'ISI'  
 PROGRAM DESCRIPTION  
 'MASTER COMMAND LIST REPORT'  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-UNCLASSIFIED  
 OUTPUT-SECURITY IS OUT-CONFIDENTIAL  
 WITHIN SUBSYSTEM SECSS  
 OUTPUT-MODE IS REPORT-MODE  
 FREQUENCY IS ADHOC  
 SERVICE-PRIORITY IS NORMAL-PRIORITY  
 RESPONSE-TIME IS OVERNIGHT-RESPONSE  
 MODE IS BATCH  
 POINT-OF-CONTACT IS 'CHIEF ROWE'  
 POC-PHONE IS '325-0880'  
 USER-ORGANIZATION IS 'NIC-44'  
 SERVICE-HISTORY IS EXISTING-SERVICE  
 COMMENTS  
 'THE NIPSSA COMMAND ALPHA LIST REPORT CAN BE PRODUCED  
 - 'FOR ONLY THOSE INCUMBENTS WITH NIPSSA BILLETS. NIPSSA  
 - 'BILLETS ARE DEFINED BY A SERIES OF COMMAND CODE/BILLET  
 - 'NUMBER RANGES. THOSE RANGES ARE: 223000/000 THROUGH  
 - '223000/999, 280232/000 THROUGH 280232/999, AND 280290/000  
 - 'THROUGH 280290/999. THE REPORT IS SORTED IN ALPHABETIC  
 - 'ORDER BY NAME. NIPSSA-015 IS THE RECIPIENT OF THIS  
 - 'REPORT.''  
 ADD PROGRAM NAME IS S8152005  
 PREPARED BY 'ISI'  
 PROGRAM DESCRIPTION  
 'MASTER COMMAND ALPHA LIST REPORT'  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-UNCLASSIFIED  
 OUTPUT-SECURITY IS OUT-CONFIDENTIAL  
 WITHIN SUBSYSTEM SECSS  
 OUTPUT-MODE IS REPORT-MODE  
 FREQUENCY IS ADHOC  
 SERVICE-PRIORITY IS NORMAL-PRIORITY  
 RESPONSE-TIME IS OVERNIGHT-RESPONSE  
 MODE IS BATCH  
 POINT-OF-CONTACT IS 'CHIEF ROWE'  
 POC-PHONE IS '325-0880'  
 USER-ORGANIZATION IS 'NIC-44'  
 SERVICE-HISTORY IS EXISTING-SERVICE  
 COMMENTS  
 'NIPSSA BI DATE LIST REPORT. AN ALTERNATE NIPSSA REPORT'

MEMBER NAME DIASMSUB

- (USING THE SAME RANGES AS ITEM A) IS SORTED IN BACKGROUND
- INVESTIGATION DATE ORDER. NIPSSA BILLETS ARE DEFINED BY A
- SERIES OF COMMAND CODE BILLET NUMBER RANGES. THOSE RANGES
- ARE: 223000/000 THROUGH 223000/999, 280232/000 THROUGH
- 280232/999, AND 280290/000 THROUGH 280290/999. NIPSSA-01S
- IS THE RECIPIENT
- OF THIS REPORT..

ADD PROGRAM NAME IS S8152006

PREPARED BY 'ISI'

PROGRAM DESCRIPTION

'MASTER COMMAND ALPHA LIST REPORT'

ENTRY-SECURITY IS ENTRY-UNCLASSIFIED

DATA-SECURITY IS DATA-UNCLASSIFIED

OUTPUT-SECURITY IS OUT-CONFIDENTIAL

WITHIN SUBSYSTEM SECSS

OUTPUT-MODE IS REPORT-MODE

FREQUENCY IS ADHOC

SERVICE-PRIORITY IS NORMAL-PRIORITY

RESPONSE-TIME IS OVERNIGHT-RESPONSE

MODE IS BATCH

POINT-OF-CONTACT IS 'CHIEF ROWE'

POC-PHONE IS '325-0880'

USER-ORGANIZATION IS 'NIC-44'

SERVICE-HISTORY IS EXISTING-SERVICE

COMMENTS

'SERIES 800XXX COMMAND ALPHA LIST. A MASTER COMMAND ALPHA

- LIST - SERIES 800XXX (PREVIOUS JOB ID #34). SELECTS FOR

- THE 800000-800999 COMMAND CODE SERIES AND LISTS IN

- ALPHABETIC ORDER BY NAME..

ADD PROGRAM NAME IS S8152007

PREPARED BY 'ISI'

PROGRAM DESCRIPTION

'MASTER COMMAND ALPHA LIST REPORT'

ENTRY-SECURITY IS ENTRY-UNCLASSIFIED

DATA-SECURITY IS DATA-UNCLASSIFIED

OUTPUT-SECURITY IS OUT-CONFIDENTIAL

WITHIN SUBSYSTEM SECSS

OUTPUT-MODE IS REPORT-MODE

FREQUENCY IS ADHOC

SERVICE-PRIORITY IS NORMAL-PRIORITY

RESPONSE-TIME IS OVERNIGHT-RESPONSE

MODE IS BATCH

POINT-OF-CONTACT IS 'CHIEF ROWE'

POC-PHONE IS '325-0880'

USER-ORGANIZATION IS 'NIC-44'

SERVICE-HISTORY IS EXISTING-SERVICE

COMMENTS

'MASTER COMMAND LIST - SERIES 800XXX. A MASTER COMMAND

- LIST - SERIES 800XXX (PREVIOUS JOB ID #35). SELECTS

- THE 800000-800999 COMMAND CODE SERIES LISTED IN COMMAND

- CODE/BILLET NUMBER ORDER..

ADD PROGRAM NAME IS S8152008

PREPARED BY 'ISI'

PROGRAM DESCRIPTION

'MASTER COMMAND ALPHA LIST REPORT'

ENTRY-SECURITY IS ENTRY-UNCLASSIFIED

DATA-SECURITY IS DATA-UNCLASSIFIED

OUTPUT-SECURITY IS OUT-CONFIDENTIAL

MEMBER NAME DIASMSUB  
 WITHIN SUBSYSTEM SECSS  
~~OUTPUT-MODE IS REPORT-MODE~~  
 FREQUENCY IS ADHOC  
 SERVICE-PRIORITY IS NORMAL-PRIORITY  
 RESPONSE-TIME IS OVERNIGHT-RESPONSE  
~~MODE IS BATCH~~  
 POINT-OF-CONTACT IS 'CHIEF ROWE'  
 POC-PHONE IS '325-0880'  
 USER-ORGANIZATION IS 'NIC-44'  
 SERVICE-HISTORY IS EXISTING-SERVICE  
 COMMENTS  
 - 'SELECTIVE COMMAND LIST REPORT. A MASTER COMMAND LIST  
 - 'REPORT (PREVIOUS JOB ID #39). SELECTS A RANGE OF  
 - 'COMMAND CODES AS SPECIFIED BY THE USER (WHICH MAY BE  
 - 'THE TOTAL FILE) AND LISTS THEM IN COMMAND CODE/BILLET  
 - 'NUMBER SEQUENCE WITH PAGE BREAKS WHEN THE COMMAND CODE  
 - 'CHANGES.'.  
 ADD PROGRAM NAME IS S8152009  
 PREPARED BY 'ISI'  
 PROGRAM DESCRIPTION  
 'NAVSIM COMMAND ALPHA LIST REPORT'  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-UNCLASSIFIED  
 OUTPUT-SECURITY IS OUT-CONFIDENTIAL  
 WITHIN SUBSYSTEM SECSS  
~~OUTPUT-MODE IS REPORT-MODE~~  
 FREQUENCY IS ADHOC  
 SERVICE-PRIORITY IS NORMAL-PRIORITY  
 RESPONSE-TIME IS OVERNIGHT-RESPONSE  
~~MODE IS BATCH~~  
 POINT-OF-CONTACT IS 'CHIEF ROWE'  
 POC-PHONE IS '325-0880'  
 USER-ORGANIZATION IS 'NIC-44'  
 SERVICE-HISTORY IS EXISTING-SERVICE  
 COMMENTS  
 - 'THE NAVSIM COMMAND ALPHA LIST REPORT (PREVIOUS  
 - 'JOB ID #32) CONTAINS ALL INCUMBENTS IN THE DATABASE  
 - 'WHO POSSESS AN, AX, ES, EX, CR, CX, CY,  
 - 'CL, MA, DN, PW, PC, OR PM ACCESSSES. LISTED  
 - 'IN ALPHABETIC ORDER BY NAME.'.  
 ADD PROGRAM NAME IS S8152010  
 PREPARED BY 'ISI'  
 PROGRAM DESCRIPTION  
 'NAVSIM SSO ADMIN REPORT'  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-UNCLASSIFIED  
 OUTPUT-SECURITY IS OUT-CONFIDENTIAL  
 WITHIN SUBSYSTEM SECSS  
~~OUTPUT-MODE IS REPORT-MODE~~  
~~FREQUENCY IS ADHOC~~  
 SERVICE-PRIORITY IS NORMAL-PRIORITY  
 RESPONSE-TIME IS OVERNIGHT-RESPONSE  
~~MODE IS BATCH~~  
~~POINT-OF-CONTACT IS 'CHIEF ROWE'~~  
~~POC-PHONE IS '325-0880'~~  
~~USER-ORGANIZATION IS 'NIC-44'~~  
~~SERVICE-HISTORY IS EXISTING-SERVICE~~  
~~COMMENTS~~



MEMBER NAME DIASMSUB

'THE NAVSIM SSO ADMIN REPORT (PREVIOUS JOB ID #40)

'CONTAINS ALL INCUMBENTS IN COMMAND CODES 010000

'THROUGH 100000 OR 200000 THROUGH 999999

'WHO POSSESS AN, AX, ES, EX, CR, CX, CY,

'CL, MA, DN, PW, PC, OR PM ACCESSES,

'LISTED IN COMMAND CODE/BILLET NUMBER

'ORDER..'

ADD PROGRAM NAME IS S8152011

PREPARED BY 'ISI'

PROGRAM DESCRIPTION

'SI COMMAND ALPHA LIST REPORT'

ENTRY-SECURITY IS ENTRY-UNCLASSIFIED

DATA-SECURITY IS DATA-UNCLASSIFIED

OUTPUT-SECURITY IS OUT-CONFIDENTIAL

WITHIN SUBSYSTEM SECSS

OUTPUT-MODE IS REPORT-MODE

FREQUENCY IS ADHOC

SERVICE-PRIORITY IS NORMAL-PRIORITY

RESPONSE-TIME IS OVERNIGHT-RESPONSE

MODE IS BATCH

POINT-OF-CONTACT IS 'CHIEF ROWE'

POC-PHONE IS '325-0880'

USER-ORGANIZATION IS 'NIC-44'

SERVICE-HISTORY IS EXISTING-SERVICE

COMMENTS

'THE SI COMMAND ALPHA LIST REPORT (PREVIOUS JOB

'ID #33) CONTAINS ALL INCUMBENTS IN THE

'DATABASE WHO POSSESS SI ACCESSES,

'LISTED IN ALPHABETIC

'ORDER BY NAME..'

ADD PROGRAM NAME IS S8152012

PREPARED BY 'ISI'

PROGRAM DESCRIPTION

'OLD SI COMMAND ALPHA LIST REPORT'

ENTRY-SECURITY IS ENTRY-UNCLASSIFIED

DATA-SECURITY IS DATA-UNCLASSIFIED

OUTPUT-SECURITY IS OUT-CONFIDENTIAL

WITHIN SUBSYSTEM SECSS

OUTPUT-MODE IS REPORT-MODE

FREQUENCY IS ADHOC

SERVICE-PRIORITY IS NORMAL-PRIORITY

RESPONSE-TIME IS OVERNIGHT-RESPONSE

MODE IS BATCH

POINT-OF-CONTACT IS 'CHIEF ROWE'

POC-PHONE IS '325-0880'

USER-ORGANIZATION IS 'NIC-44'

SERVICE-HISTORY IS EXISTING-SERVICE

COMMENTS

'THE OLD SI COMMAND ALPHA LIST REPORT (PREVIOUS JOB ID #36)

'CONTAINS ALL INCUMBENTS WITH NAVY COMMAND CODES (200000-

'299999 OR 700000-799999) IN THE DATABASE WHO POSSESS

'SI ACCESSES MORE THAN 4 1/2 YEARS OLD, LISTED IN

'ALPHABETIC ORDER BY NAME..'

ADD PROGRAM NAME IS S8152013

PREPARED BY 'ISI'

PROGRAM DESCRIPTION

'OLD SI COMMAND LIST REPORT'

ENTRY-SECURITY IS ENTRY-UNCLASSIFIED



MEMBER NAME DIASMSUB

DATA-SECURITY IS DATA-UNCLASSIFIED  
OUTPUT-SECURITY IS OUT-CONFIDENTIAL  
WITHIN SUBSYSTEM SECSS  
OUTPUT-MODE IS REPORT-MODE  
FREQUENCY IS ADHOC  
SERVICE-PRIORITY IS NORMAL-PRIORITY  
RESPONSE-TIME IS OVERNIGHT-RESPONSE  
MODE IS BATCH  
POINT-OF-CONTACT IS 'CHIEF ROWE'  
POC-PHONE IS '325-0880'  
USER-ORGANIZATION IS 'NIC-44'  
SERVICE-HISTORY IS EXISTING-SERVICE  
COMMENTS

- 'THE OLD BI COMMAND LIST REPORT (PREVIOUS JOB ID #42)
- 'CONTAINS ALL INCUMBENTS WITH NAVY COMMAND CODES (200000-
- '299999 OR 700000-799999) IN THE DATABASE WHO POSSESS
- 'SI ACCESSES MORE THAN 4 1/2 YEARS OLD, LISTED IN
- 'COMMAND CODE/BILLET NUMBER SEQUENCE..'

ADD PROGRAM NAME IS S8152014

PREPARED BY 'ISI'

PROGRAM DESCRIPTION

'MASTER COMMAND BILLET LIST REPORT'

ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
DATA-SECURITY IS DATA-UNCLASSIFIED  
OUTPUT-SECURITY IS OUT-CONFIDENTIAL  
WITHIN SUBSYSTEM SECSS  
OUTPUT-MODE IS REPORT-MODE  
FREQUENCY IS ADHOC  
SERVICE-PRIORITY IS NORMAL-PRIORITY  
RESPONSE-TIME IS OVERNIGHT-RESPONSE  
MODE IS BATCH  
POINT-OF-CONTACT IS 'CHIEF ROWE'  
POC-PHONE IS '325-0880'  
USER-ORGANIZATION IS 'NIC-44'  
SERVICE-HISTORY IS EXISTING-SERVICE  
COMMENTS

- 'THE MASTER COMMAND BILLET LIST (PREVIOUS JOB ID #43)
- 'CONTAINS ALL BILLETS IN ALL COMMANDS IN THE DATABASE,
- 'LISTED IN COMMAND CODE/BILLET NUMBER ORDER..'

ADD PROGRAM NAME IS S8152015

PREPARED BY 'ISI'

PROGRAM DESCRIPTION

'MASTER COMMAND TITLES/BILLET REPORT'

ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
DATA-SECURITY IS DATA-UNCLASSIFIED  
OUTPUT-SECURITY IS OUT-CONFIDENTIAL  
WITHIN SUBSYSTEM SECSS  
OUTPUT-MODE IS REPORT-MODE  
FREQUENCY IS ADHOC  
SERVICE-PRIORITY IS NORMAL-PRIORITY  
RESPONSE-TIME IS OVERNIGHT-RESPONSE  
MODE IS BATCH  
POINT-OF-CONTACT IS 'CHIEF ROWE'  
POC-PHONE IS '325-0880'  
USER-ORGANIZATION IS 'NIC-44'  
SERVICE-HISTORY IS EXISTING-SERVICE  
COMMENTS

'THE MASTER COMMAND TITLES/BILLET REPORT (PREVIOUS

MEMBER NAME DIASMSUB

- 'JOB ID #46) IS FOR ALL COMMANDS ON FILE, LISTED IN
- 'COMMAND CODE/BILLET NUMBER ORDER..'
- ADD PROGRAM NAME IS S8152016
- PREPARED BY 'ISI'
- PROGRAM DESCRIPTION
- 'MASTER ALPHA COMMAND TITLES/BILLET REPORT'
- ENTRY-SECURITY IS ENTRY-UNCLASSIFIED
- DATA-SECURITY IS DATA-UNCLASSIFIED
- OUTPUT-SECURITY IS OUT-CONFIDENTIAL
- WITHIN SUBSYSTEM SECSS
- OUTPUT-MODE IS REPORT-MODE
- FREQUENCY IS ADHOC
- SERVICE-PRIORITY IS NORMAL-PRIORITY
- RESPONSE-TIME IS OVERNIGHT-RESPONSE
- MODE IS BATCH
- POINT-OF-CONTACT IS 'CHIEF ROWE'
- POC-PHONE IS '325-0880'
- USER-ORGANIZATION IS 'NIC-44'
- SERVICE-HISTORY IS EXISTING-SERVICE
- COMMENTS
- 'THE MASTER ALPHA COMMAND TITLES/BILLET REPORT (PREVIOUS
- 'JOB ID #49) IS FOR ALL COMMANDS ON FILE, LISTED IN
- 'ALPHABETIC ORDER BY NAME..'
- ADD PROGRAM NAME IS S8152017
- PREPARED BY 'ISI'
- PROGRAM DESCRIPTION
- 'MASTER COMMAND TOTALS - INCUMBENT NAVY'
- ENTRY-SECURITY IS ENTRY-UNCLASSIFIED
- DATA-SECURITY IS DATA-UNCLASSIFIED
- OUTPUT-SECURITY IS OUT-CONFIDENTIAL
- WITHIN SUBSYSTEM SECSS
- OUTPUT-MODE IS REPORT-MODE
- FREQUENCY IS ADHOC
- SERVICE-PRIORITY IS NORMAL-PRIORITY
- RESPONSE-TIME IS OVERNIGHT-RESPONSE
- MODE IS BATCH
- POINT-OF-CONTACT IS 'CHIEF ROWE'
- POC-PHONE IS '325-0880'
- USER-ORGANIZATION IS 'NIC-44'
- SERVICE-HISTORY IS EXISTING-SERVICE
- COMMENTS
- 'MASTER COMMAND TOTALS - INCUMBENT NAVY ONLY (PREVIOUS
- 'JOB ID #47). THIS REPORT CONTAINS TOTALS OF ALL
- 'ACCESSES (EXCLUDING Z TYPE) ASSIGNED TO INCUMBENTS, SORTED
- 'IN COMMAND CODE SEQUENCE, SUBTOTALLED BY COMMAND CODE,
- 'AND GRAND TOTALLED FOR ALL NAVY COMMANDS, ASSIGNED COMMAND
- 'CODES FOR THE NAVY ARE IN THE FOLLOWING RANGES:
- '200000 THROUGH 299999, 722000 THROUGH 722999,
- '732000 THROUGH 732999, 742000 THROUGH 742999,
- '772000 THROUGH 772999, 774000 THROUGH 774999, AND
- '782000 THROUGH 782999..'
- ADD PROGRAM NAME IS S8152018
- PREPARED BY 'ISI'
- PROGRAM DESCRIPTION
- 'MASTER COMMAND TOTALS - INCUMBENT NON-NAVY'
- ENTRY-SECURITY IS ENTRY-UNCLASSIFIED
- DATA-SECURITY IS DATA-UNCLASSIFIED
- OUTPUT-SECURITY IS OUT-CONFIDENTIAL

MEMBER NAME DIASMSUM

WITHIN SUBSYSTEM SECSS

OUTPUT-MODE IS REPORT-MODE

FREQUENCY IS ADHOC

SERVICE-PRIORITY IS NORMAL-PRIORITY

RESPONSE-TIME IS OVERNIGHT-RESPONSE

MODE IS BATCH

POINT-OF-CONTACT IS 'CHIEF ROWE'

POC-PHONE IS '325-0880'

USER-ORGANIZATION IS 'NIC-44'

SERVICE-HISTORY IS EXISTING-SERVICE

COMMENTS

- 'MASTER COMMAND TOTALS - INCUMBENT NON-NAVY ONLY (PREVIOUS
- 'JOB ID #47). THIS REPORT CONTAINS TOTALS OF ALL
- 'ACCESSES (EXCLUDING Z-TYPE) ASSIGNED TO INCUMBENTS, SORTED
- 'IN COMMAND CODE SEQUENCE, SUBTOTALLED BY COMMAND CODE,
- 'AND GRAND TOTALLED FOR ALL NON-NAVY COMMANDS. ASSIGNED
- 'COMMAND CODES FOR THE NAVY (EXCLUDED IN THIS LISTING) ARE
- 'IN THE FOLLOWING RANGES:
- '200000 THROUGH 299999, 722000 THROUGH 722999,
- '732000 THROUGH 732999, 742000 THROUGH 742999,
- '772000 THROUGH 772999, 774000 THROUGH 774999, AND
- '782000 THROUGH 782999,
- 'COMMAND CODES 800000 THROUGH 899999 ARE ALSO EXCLUDED..

ADD PROGRAM NAME IS S8152019

PREPARED BY 'ISI'

PROGRAM DESCRIPTION

'MASTER COMMAND TOTALS - INCUMBENT Z DATA'

ENTRY-SECURITY IS ENTRY-UNCLASSIFIED

DATA-SECURITY IS DATA-UNCLASSIFIED

OUTPUT-SECURITY IS OUT-CONFIDENTIAL

WITHIN SUBSYSTEM SECSS

OUTPUT-MODE IS REPORT-MODE

FREQUENCY IS ADHOC

SERVICE-PRIORITY IS NORMAL-PRIORITY

RESPONSE-TIME IS OVERNIGHT-RESPONSE

MODE IS BATCH

POINT-OF-CONTACT IS 'CHIEF ROWE'

POC-PHONE IS '325-0880'

USER-ORGANIZATION IS 'NIC-44'

SERVICE-HISTORY IS EXISTING-SERVICE

COMMENTS

- 'MASTER COMMAND TOTALS - INCUMBENT Z DATA (PREVIOUS
- 'JOB ID #47). THIS REPORT CONTAINS TOTALS OF ALL Z
- 'ACCESSES ASSIGNED TO INCUMBENTS, SORTED IN COMMAND CODE
- 'SEQUENCE, SUBTOTALLED BY COMMAND CODE, AND GRAND
- 'TOTALLED FOR ALL COMMAND CODES..

ADD PROGRAM NAME IS S8152020

PREPARED BY 'ISI'

PROGRAM DESCRIPTION

'MASTER COMMAND TOTALS - 800000 SERIES'

ENTRY-SECURITY IS ENTRY-UNCLASSIFIED

DATA-SECURITY IS DATA-UNCLASSIFIED

OUTPUT-SECURITY IS OUT-CONFIDENTIAL

WITHIN SUBSYSTEM SECSS

OUTPUT-MODE IS REPORT-MODE

FREQUENCY IS ADHOC

SERVICE-PRIORITY IS NORMAL-PRIORITY

RESPONSE-TIME IS OVERNIGHT-RESPONSE

MEMBER NAME DIASMSUM

MODE IS BATCH

POINT-OF-CONTACT IS CHIEF ROWE

POC-PHONE IS 325-0880

USER-ORGANIZATION IS NIC-44

SERVICE-HISTORY IS EXISTING-SERVICE

COMMENTS

- MASTER COMMAND TOTALS - 800000 SERIES (PREVIOUS
- JOB ID #47). THIS REPORT CONTAINS TOTALS OF ALL
- ACCESSES (EXCLUDING Z TYPE) ASSIGNED TO INCUMBENTS, SORTED
- IN COMMAND CODE SEQUENCE, SUBTOTALLED BY COMMAND CODE,
- AND GRAND TOTALLED FOR ALL COMMAND CODES IN THE RANGE
- 800000 THROUGH 899999.

ADD PROGRAM NAME IS SH152021

PREPARED BY ISI

PROGRAM DESCRIPTION

MASTER COMMAND TOTALS - BILLET NAVY ONLY

ENTRY-SECURITY IS ENTRY-UNCLASSIFIED

DATA-SECURITY IS DATA-UNCLASSIFIED

OUTPUT-SECURITY IS OUT-CONFIDENTIAL

WITHIN SUBSYSTEM SECSS

OUTPUT-MODE IS REPORT-MODE

FREQUENCY IS ADHOC

SERVICE-PRIORITY IS NORMAL-PRIORITY

RESPONSE-TIME IS OVERNIGHT-RESPONSE

MODE IS BATCH

POINT-OF-CONTACT IS CHIEF ROWE

POC-PHONE IS 325-0880

USER-ORGANIZATION IS NIC-44

SERVICE-HISTORY IS EXISTING-SERVICE

COMMENTS

- MASTER COMMAND TOTALS - BILLET NAVY ONLY (PREVIOUS
- JOB ID #48). THIS REPORT CONTAINS TOTALS OF ALL
- ACCESSES (EXCLUDING Z TYPE) ASSIGNED TO BILLETS, SORTED
- IN COMMAND CODE SEQUENCE, SUBTOTALLED BY COMMAND CODE,
- AND GRAND TOTALLED FOR ALL NAVY COMMANDS. ASSIGNED COMMAND
- CODES FOR THE NAVY ARE IN THE FOLLOWING RANGES:
- 200000 THROUGH 299999, 722000 THROUGH 722999,
- 732000 THROUGH 732999, 742000 THROUGH 742999,
- 772000 THROUGH 772999, 774000 THROUGH 774999, AND
- 782000 THROUGH 782999.

ADD PROGRAM NAME IS SH152022

PREPARED BY ISI

PROGRAM DESCRIPTION

MASTER COMMAND TOTALS - BILLET NON-NAVY

ENTRY-SECURITY IS ENTRY-UNCLASSIFIED

DATA-SECURITY IS DATA-UNCLASSIFIED

OUTPUT-SECURITY IS OUT-CONFIDENTIAL

WITHIN SUBSYSTEM SECSS

OUTPUT-MODE IS REPORT-MODE

FREQUENCY IS ADHOC

SERVICE-PRIORITY IS NORMAL-PRIORITY

RESPONSE-TIME IS OVERNIGHT-RESPONSE

MODE IS BATCH

POINT-OF-CONTACT IS CHIEF ROWE

POC-PHONE IS 325-0880

USER-ORGANIZATION IS NIC-44

SERVICE-HISTORY IS EXISTING-SERVICE

COMMENTS



MEMBER NAME DIASMSUB

- MASTER COMMAND TOTALS - BILLET NON-NAVY ONLY (PREVIOUS
- JOB ID #48). THIS REPORT CONTAINS TOTALS OF ALL
- ACCESSSES (EXCLUDING Z TYPE) ASSIGNED TO BILLETS, SORTED
- IN COMMAND CODE SEQUENCE, SUBTOTALLED BY COMMAND CODE,
- AND GRAND TOTALLED FOR ALL NON-NAVY COMMANDS. ASSIGNED
- COMMAND CODES FOR THE NAVY (EXCLUDED IN THIS LISTING) ARE
- IN THE FOLLOWING RANGES:
- 200000 THROUGH 299999, 722000 THROUGH 722999,
- 732000 THROUGH 732999, 742000 THROUGH 742999,
- 772000 THROUGH 772999, 774000 THROUGH 774999, AND
- 782000 THROUGH 782999.
- COMMAND CODES 800000 THROUGH 899999 ARE ALSO EXCLUDED..

ADD PROGRAM NAME IS 58152023

PREPARED BY ISI

PROGRAM DESCRIPTION

MASTER COMMAND TOTALS - 800000 SERIES

ENTRY-SECURITY IS ENTRY-UNCLASSIFIED

DATA-SECURITY IS DATA-UNCLASSIFIED

OUTPUT-SECURITY IS OUT-CONFIDENTIAL

WITHIN SUBSYSTEM SECSS

OUTPUT-MODE IS REPORT-MODE

FREQUENCY IS ADHOC

SERVICE-PRIORITY IS NORMAL-PRIORITY

RESPONSE-TIME IS OVERNIGHT-RESPONSE

MODE IS BATCH

POINT-OF-CONTACT IS CHIEF ROWE

POC-PHONE IS 325-0880

USER-ORGANIZATION IS NIC-44

SERVICE-HISTORY IS EXISTING-SERVICE

COMMENTS

- MASTER COMMAND TOTALS - 800000 SERIES (PREVIOUS
- JOB ID #48). THIS REPORT CONTAINS TOTALS OF ALL
- ACCESSSES (EXCLUDING Z TYPE) ASSIGNED TO BILLETS, SORTED
- IN COMMAND CODE SEQUENCE, SUBTOTALLED BY COMMAND CODE,
- AND GRAND TOTALLED FOR ALL COMMAND CODES IN THE RANGE
- 800000 THROUGH 899999..

ADD PROGRAM NAME IS 58152024

PREPARED BY ISI

PROGRAM DESCRIPTION

MASTER COMMAND TOTALS - BILLET Z DATA

ENTRY-SECURITY IS ENTRY-UNCLASSIFIED

DATA-SECURITY IS DATA-UNCLASSIFIED

OUTPUT-SECURITY IS OUT-CONFIDENTIAL

WITHIN SUBSYSTEM SECSS

OUTPUT-MODE IS REPORT-MODE

FREQUENCY IS ADHOC

SERVICE-PRIORITY IS NORMAL-PRIORITY

RESPONSE-TIME IS OVERNIGHT-RESPONSE

MODE IS BATCH

POINT-OF-CONTACT IS CHIEF ROWE

POC-PHONE IS 325-0880

USER-ORGANIZATION IS NIC-44

SERVICE-HISTORY IS EXISTING-SERVICE

COMMENTS

- MASTER COMMAND TOTALS - BILLET Z DATA (PREVIOUS
- JOB ID #48). THIS REPORT CONTAINS TOTALS OF ALL Z
- ACCESSSES ASSIGNED TO BILLETS, SORTED IN COMMAND CODE
- SEQUENCE, SUBTOTALLED BY COMMAND CODE, AND GRAND



MEMBER NAME DIASMSUB

- TOTALED FOR ALL COMMAND CODES..
- ADD PROGRAM NAME IS S4152025
- PREPARED BY 'ISI'
- PROGRAM DESCRIPTION
- 'MASTER SOCIAL SECURITY LIST REPORT'
- ENTRY-SECURITY IS ENTRY-UNCLASSIFIED
- DATA-SECURITY IS DATA-UNCLASSIFIED
- OUTPUT-SECURITY IS OUT-CONFIDENTIAL
- WITHIN SUBSYSTEM SECSS
- OUTPUT-MODE IS REPORT-MODE
- FREQUENCY IS ADHOC
- SERVICE-PRIORITY IS NORMAL-PRIORITY
- RESPONSE-TIME IS OVERNIGHT-RESPONSE
- MODE IS BATCH
- POINT-OF-CONTACT IS 'CHIEF ROWE'
- POC-PHONE IS '325-0880'
- USER-ORGANIZATION IS 'NIC-44'
- SERVICE-HISTORY IS EXISTING-SERVICE
- COMMENTS
- 'THE MASTER SOCIAL SECURITY LIST (PREVIOUS JOB ID #55)
- 'REPORT CONTAINS ALL INCUMBENTS IN THE DATABASE.
- 'LISTED IN SOCIAL SECURITY NUMBER ORDER..'
- ADD PROGRAM NAME IS S4152026
- PREPARED BY 'ISI'
- PROGRAM DESCRIPTION
- 'Z COMMAND ALPHA LIST REPORT'
- ENTRY-SECURITY IS ENTRY-UNCLASSIFIED
- DATA-SECURITY IS DATA-UNCLASSIFIED
- OUTPUT-SECURITY IS OUT-CONFIDENTIAL
- WITHIN SUBSYSTEM SECSS
- OUTPUT-MODE IS REPORT-MODE
- FREQUENCY IS ADHOC
- SERVICE-PRIORITY IS NORMAL-PRIORITY
- RESPONSE-TIME IS OVERNIGHT-RESPONSE
- MODE IS BATCH
- POINT-OF-CONTACT IS 'CHIEF ROWE'
- POC-PHONE IS '325-0880'
- USER-ORGANIZATION IS 'NIC-44'
- SERVICE-HISTORY IS EXISTING-SERVICE
- COMMENTS
- 'THE Z COMMAND ALPHA LIST REPORT (PREVIOUS JOB ID #56)
- 'CONTAINS ALL INCUMBENTS IN THE
- 'DATABASE WHO POSSESS Z OR MED ACCESSSES
- 'LISTED IN ALPHABETIC ORDER BY NAME..'
- ADD PROGRAM NAME IS S4152027
- PREPARED BY 'ISI'
- PROGRAM DESCRIPTION
- 'Z COMMAND ALPHA LIST REPORT - MED ONLY'
- ENTRY-SECURITY IS ENTRY-UNCLASSIFIED
- DATA-SECURITY IS DATA-UNCLASSIFIED
- OUTPUT-SECURITY IS OUT-CONFIDENTIAL
- WITHIN SUBSYSTEM SECSS
- OUTPUT-MODE IS REPORT-MODE
- FREQUENCY IS ADHOC
- SERVICE-PRIORITY IS NORMAL-PRIORITY
- RESPONSE-TIME IS OVERNIGHT-RESPONSE
- MODE IS BATCH
- POINT-OF-CONTACT IS 'CHIEF ROWE'

AD-A069 868

NAVAL INTELLIGENCE PROCESSING SYSTEMS SUPPORT ACTIVITIES--ETC F/6 9/2  
INTEGRATED DATABASE DEVELOPMENT AND DESIGN GUIDE. VERSION 2.0.(U)  
MAY 79

UNCLASSIFIED

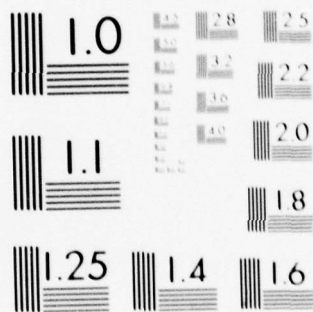
NIPSSA-UM-8000/2.0-5/79

NL

6 OF 7

AD  
A069868





MICROCOPY RESOLUTION TEST CHART  
NATIONAL BUREAU OF STANDARDS-1963-A

MEMBER NAME DIASMSUM  
 POC-PHONE IS '325-0880'  
 USER-ORGANIZATION IS 'NIC-44'  
 SERVICE-HISTORY IS EXISTING-SERVICE  
 COMMENTS  
 - 'THE Z COMMAND ALPHA LIST REPORT - MED ONLY (PREVIOUS  
 - 'JOB ID #62) CONTAINS ALL INCUMBENTS IN THE  
 - 'DATABASE WHO POSSESS A MED ACCESS  
 - 'LISTED IN ALPHABETIC ORDER BY NAME..'  
 ADD) PROGRAM NAME IS SB152028  
 PREPARED BY 'ISI'  
 PROGRAM DESCRIPTION  
 'Z COMMAND ALPHA LIST REPORT - DUL ONLY'  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-UNCLASSIFIED  
 OUTPUT-SECURITY IS OUT-CONFIDENTIAL  
 WITHIN SUBSYSTEM SECSS  
 OUTPUT-MODE IS REPORT-MODE  
 FREQUENCY IS ADHOC  
 SERVICE-PRIORITY IS NORMAL-PRIORITY  
 RESPONSE-TIME IS OVERNIGHT-RESPONSE  
 MODE IS BATCH  
 POINT-OF-CONTACT IS 'CHIEF ROWE'  
 POC-PHONE IS '325-0880'  
 USER-ORGANIZATION IS 'NIC-44'  
 SERVICE-HISTORY IS EXISTING-SERVICE  
 COMMENTS  
 - 'THE Z COMMAND ALPHA LIST REPORT - DUL ONLY (PREVIOUS  
 - 'JOB ID #53) CONTAINS ALL INCUMBENTS IN THE  
 - 'DATABASE WHO POSSESS DUL ACCESSES LISTED IN  
 - 'ALPHABETIC ORDER BY NAME..'  
 ADD) PROGRAM NAME IS SB152029  
 PREPARED BY 'ISI'  
 PROGRAM DESCRIPTION  
 'THE Z COMMAND LIST REPORT'  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-UNCLASSIFIED  
 OUTPUT-SECURITY IS OUT-CONFIDENTIAL  
 WITHIN SUBSYSTEM SECSS  
 OUTPUT-MODE IS REPORT-MODE  
 FREQUENCY IS ADHOC  
 SERVICE-PRIORITY IS NORMAL-PRIORITY  
 RESPONSE-TIME IS OVERNIGHT-RESPONSE  
 MODE IS BATCH  
 POINT-OF-CONTACT IS 'CHIEF ROWE'  
 POC-PHONE IS '325-0880'  
 USER-ORGANIZATION IS 'NIC-44'  
 SERVICE-HISTORY IS EXISTING-SERVICE  
 COMMENTS  
 - 'THE Z COMMAND LIST REPORT (PREVIOUS JOB ID #57)  
 - 'CONTAINS ALL INCUMBENTS IN THE  
 - 'DATABASE WHO POSSESS Z OR MED ACCESSSES  
 - 'LISTED IN COMMAND CODE/BILLET NUMBER ORDER. WITH  
 - 'A PAGE BREAK WHEN THE COMMAND CODE CHANGES..'  
 ADD) PROGRAM NAME IS SB152030  
 PREPARED BY 'ISI'  
 PROGRAM DESCRIPTION  
 'Z COMMAND LIST REPORT - MED ACCESSSES ONLY'  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED

MEMBER NAME DIASMSUB  
 DATA-SECURITY IS DATA-UNCLASSIFIED  
 OUTPUT-SECURITY IS OUT-CONFIDENTIAL  
 WITHIN SUBSYSTEM SECSS  
 OUTPUT-MODE IS REPORT-MODE  
 FREQUENCY IS ADHOC  
 SERVICE-PRIORITY IS NORMAL-PRIORITY  
 RESPONSE-TIME IS OVERNIGHT-RESPONSE  
 MODE IS BATCH  
 POINT-OF-CONTACT IS 'CHIEF ROWE'  
 POC-PHONE IS '325-0880'  
 USER-ORGANIZATION IS 'NIC-44'  
 SERVICE-HISTORY IS EXISTING-SERVICE  
 COMMENTS  
 - THE 7 COMMAND LIST REPORT - MED ACCESSSES ONLY (PREVIOUS  
 - 'JOB ID #61) CONTAINS ALL INCUMBENTS IN THE  
 - 'DATABASE WHO POSSESS A MED ACCESS LISTED IN  
 - 'ALPHABETIC ORDER BY NAME WITHIN COMMAND CODE..'  
 ADD PROGRAM NAME IS SB152031  
 PREPARED BY 'ISI'  
 PROGRAM DESCRIPTION  
 'INCUMBENT B ACCESS REPORT'  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-UNCLASSIFIED  
 OUTPUT-SECURITY IS OUT-CONFIDENTIAL  
 WITHIN SUBSYSTEM SECSS  
 OUTPUT-MODE IS REPORT-MODE  
 FREQUENCY IS ADHOC  
 SERVICE-PRIORITY IS NORMAL-PRIORITY  
 RESPONSE-TIME IS OVERNIGHT-RESPONSE  
 MODE IS BATCH  
 POINT-OF-CONTACT IS 'CHIEF ROWE'  
 POC-PHONE IS '325-0880'  
 USER-ORGANIZATION IS 'NIC-44'  
 SERVICE-HISTORY IS EXISTING-SERVICE  
 COMMENTS  
 - THE INCUMBENT B ACCESS REPORT CONTAINS  
 - 'ALL INCUMBENTS IN THE DATABASE WHO  
 - 'POSSESS ANY TYPE OF B ACCESS. LISTED  
 - 'IN ALPHABETICAL SEQUENCE BY NAME..'  
 ADD PROGRAM NAME IS SB152032  
 PREPARED BY 'ISI'  
 PROGRAM DESCRIPTION  
 'BILLET B ACCESS REPORT'  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-UNCLASSIFIED  
 OUTPUT-SECURITY IS OUT-CONFIDENTIAL  
 WITHIN SUBSYSTEM SECSS  
 OUTPUT-MODE IS REPORT-MODE  
 FREQUENCY IS ADHOC  
 SERVICE-PRIORITY IS NORMAL-PRIORITY  
 RESPONSE-TIME IS OVERNIGHT-RESPONSE  
 MODE IS BATCH  
 POINT-OF-CONTACT IS 'CHIEF ROWE'  
 POC-PHONE IS '325-0880'  
 USER-ORGANIZATION IS 'NIC-44'  
 SERVICE-HISTORY IS EXISTING-SERVICE  
 COMMENTS  
 - THE BILLET B ACCESS REPORT CONTAINS



MEMBER NAME DIASMSUB

- 'ALL BILLETS IN THE DATABASE WHICH
- 'ARE ASSIGNED ANY TYPE OF R ACCESS, LISTED
- 'IN COMMAND CODE/BILLET NUMBER SEQUENCE.'

ADD PROGRAM NAME IS S8152033

PREPARED BY 'ISI'

PROGRAM DESCRIPTION

'BILLET/INCUMBENT MERGE REPORT'  
ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
DATA-SECURITY IS DATA-UNCLASSIFIED  
OUTPUT-SECURITY IS OUT-CONFIDENTIAL  
WITHIN SUBSYSTEM SECSS  
OUTPUT-MODE IS REPORT-MODE  
FREQUENCY IS ADHOC  
SERVICE-PRIORITY IS NORMAL-PRIORITY  
RESPONSE-TIME IS OVERNIGHT-RESPONSE  
MODE IS BATCH  
POINT-OF-CONTACT IS 'CHIEF ROWE'  
POC-PHONE IS '325-0880'  
USER-ORGANIZATION IS 'NIC-44'  
SERVICE-HISTORY IS EXISTING-SERVICE  
COMMENTS

- 'THE BILLET/INCUMBENT MERGE REPORT
- 'CONTAINS ALL BILLET AND INCUMBENT
- 'DATA FOR THE COMMANDS DEFINED BY
- 'THE USER SUPPLIED SELECTION PARAMETER,
- 'LISTED IN COMMAND CODE/BILLET
- 'NUMBER SEQUENCE.'

ADD PROGRAM NAME IS S8152034

PREPARED BY 'ISI'

PROGRAM DESCRIPTION

'CLEARANCE DELETION REPORT (INCUMBENT)'  
ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
DATA-SECURITY IS DATA-UNCLASSIFIED  
OUTPUT-SECURITY IS OUT-CONFIDENTIAL  
WITHIN SUBSYSTEM SECSS  
OUTPUT-MODE IS REPORT-MODE  
FREQUENCY IS ADHOC  
SERVICE-PRIORITY IS NORMAL-PRIORITY  
RESPONSE-TIME IS OVERNIGHT-RESPONSE  
MODE IS BATCH  
POINT-OF-CONTACT IS 'CHIEF ROWE'  
POC-PHONE IS '325-0880'  
USER-ORGANIZATION IS 'NIC-44'  
SERVICE-HISTORY IS EXISTING-SERVICE  
COMMENTS

- 'ALL DATA BASE RECORDS WHICH CONTAIN
- 'NO ACCESSES AFTER INDICATED ACCESSES
- 'ARE DELETED ARE LISTED ON THIS REPORT.
- 'THE CLEARANCE DELETION REPORT ALSO GIVES
- 'RECORD COUNTS.'

ADD PROGRAM NAME IS S8152035

PREPARED BY 'ISI'

PROGRAM DESCRIPTION

'DIA PUNCHED CARDS'  
ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
DATA-SECURITY IS DATA-UNCLASSIFIED  
OUTPUT-SECURITY IS OUT-CONFIDENTIAL  
WITHIN SUBSYSTEM SECSS

MEMBER NAME DIASMSUB

OUTPUT-MODE IS TAPE-MODE  
FREQUENCY IS QUARTERLY  
SERVICE-PRIORITY IS NORMAL-PRIORITY  
RESPONSE-TIME IS OVERNIGHT-RESPONSE  
MODE IS BATCH  
POINT-OF-CONTACT IS CHIEF ROWE  
POC-PHONE IS 1325-0840  
USER-ORGANIZATION IS NIC-44  
SERVICE-HISTORY IS EXISTING-SERVICE  
COMMENTS

- DIA PUNCHED CARDS ARE PUNCHED ONTO THE DIA PUNCHED
- CARD TAPE TO BE PUNCHED BY NIPSSA. DIA PUNCHED CARDS
- ARE PRODUCED AT THE REQUEST OF THE USER. ONE PUNCHED
- CARD IS PRODUCED FOR EACH COMMAND THAT BEGINS WITH
- A 2 AND HAS AN SSO COMMAND FLAG OF A. THE TAPE CONTAINS
- ONE RECORD FOR EACH HILLET IN THE DATABASE AND FOR
- EACH INCUMBENT IN THE INCUMBENT DATABASE WHOSE COMMAND
- CODE BEGINS WITH A 2 AND WHO POSSESS A TK, BR, BI, RE,
- BG, ER, BH, OR PO ACCESS. THIS JOB IS NORMALLY RUN
- QUARTERLY..

ADD PROGRAM NAME IS S8152036

PREPARED BY ISI

PROGRAM DESCRIPTION

SSO ADMINISTRATIVE REPORT  
ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
DATA-SECURITY IS DATA-UNCLASSIFIED  
OUTPUT-SECURITY IS OUT-CONFIDENTIAL  
WITHIN SUBSYSTEM SECSS  
OUTPUT-MODE IS REPORT-MODE  
FREQUENCY IS AD-HOC

SERVICE-PRIORITY IS NORMAL-PRIORITY  
RESPONSE-TIME IS OVERNIGHT-RESPONSE  
MODE IS BATCH  
POINT-OF-CONTACT IS CHIEF ROWE

POC-PHONE IS 1325-0880  
USER-ORGANIZATION IS NIC-44  
SERVICE-HISTORY IS EXISTING-SERVICE  
COMMENTS

- THE SSO ADMINISTRATIVE REPORT (PREVIOUS JOB ID #52)
- CONTAINS ALL HILLETS WITH THEIR ASSOCIATED
- INCUMBENTS AND ACCESSSES (EXCLUDING 2 ACCESSSES)
- FOR THE COMMAND CODE RANGES OF:
- 010000 THROUGH 199999.
- 219800 THROUGH 269999.
- 270500 THROUGH 799999, AND
- 991000 THROUGH 999999.
- SORTED IN SSO COMMAND CODE, COMMAND CODE, AND HILLET
- NUMBER SEQUENCE. CAUTION: THIS REPORT IS EXTREMELY
- LONG AND SHOULD BE SCHEDULED WITH THE ACP CENTER..

ADD PROGRAM NAME IS S8152037

PREPARED BY ISI

PROGRAM DESCRIPTION

SELECTIVE SSO ADMINISTRATIVE REPORTS

ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
DATA-SECURITY IS DATA-UNCLASSIFIED  
OUTPUT-SECURITY IS OUT-CONFIDENTIAL  
WITHIN SUBSYSTEM SECSS  
OUTPUT-MODE IS REPORT-MODE

MEMBER NAME DIASMSUH

FREQUENCY IS ADHOC

SERVICE-PRIORITY IS ~~NORMAL-PRIORITY~~

RESPONSE-TIME IS OVERNIGHT-RESPONSE

MODE IS BATCH

POINT-OF-CONTACT IS 'CHIEF ROWE'

POC-PHONE IS '325-0880'

USER-ORGANIZATION IS 'NIC-44'

SERVICE-HISTORY IS NEW-SERVICE

COMMENTS

- 'THE SELECTIVE SSO ADMINISTRATIVE REPORT CONTAINS ALL
- 'BILLETS WITH THEIR ASSOCIATED INCUMBENTS AND ACCESSES
- '(EXCLUDING Z ACCESSES) FOR THE SPECIFIC COMMAND CODE
- 'RANGES THAT ARE ENTERED BY THE USER.
- 'SORTED IN SSO COMMAND CODE, COMMAND CODE, AND BILLET
- 'NUMBER SEQUENCE.'

ADD PROGRAM NAME IS S8152038

PREPARED BY 'ISI'

PROGRAM DESCRIPTION

'SSO ADMINISTRATIVE Z LIST REPORT'

ENTRY-SECURITY IS ENTRY-UNCLASSIFIED

DATA-SECURITY IS DATA-UNCLASSIFIED

OUTPUT-SECURITY IS OUT-CONFIDENTIAL

WITHIN SUBSYSTEM SECSS

OUTPUT-MODE IS REPORT-MODE

FREQUENCY IS ADHOC

SERVICE-PRIORITY IS ~~NORMAL-PRIORITY~~

RESPONSE-TIME IS OVERNIGHT-RESPONSE

MODE IS BATCH

POINT-OF-CONTACT IS 'CHIEF ROWE'

POC-PHONE IS '325-0880'

USER-ORGANIZATION IS 'NIC-44'

SERVICE-HISTORY IS EXISTING-SERVICE

COMMENTS

- 'THE SSO ADMINISTRATIVE Z LIST REPORT CONTAINS
- 'ALL BILLETS WITH THEIR ASSOCIATED INCUMBENTS AND
- 'ACCESSES (ONLY Z ACCESSES) FOR THE COMMAND CODES
- 'SPECIFIED BELOW. CAUTION: THIS REPORT IS EXTREMELY
- 'LONG AND SHOULD BE SCHEDULED WITH THE ADD CENTER.
- '010000 THROUGH 199999, 219800 THROUGH 269999,
- '270500 THROUGH 789999, AND 991000 THROUGH 999000
- 'SORTED IN SSO COMMAND CODE, COMMAND CODE, AND
- 'BILLET NUMBER SEQUENCE.'

ADD PROGRAM NAME IS S8152039

PREPARED BY 'ISI'

PROGRAM DESCRIPTION

'NIPSSA ADMINISTRATIVE REPORT'

ENTRY-SECURITY IS ENTRY-UNCLASSIFIED

DATA-SECURITY IS DATA-UNCLASSIFIED

OUTPUT-SECURITY IS OUT-CONFIDENTIAL

WITHIN SUBSYSTEM SECSS

OUTPUT-MODE IS REPORT-MODE

FREQUENCY IS ADHOC

SERVICE-PRIORITY IS ~~NORMAL-PRIORITY~~

RESPONSE-TIME IS ~~OVERNIGHT-RESPONSE~~

MODE IS BATCH

POINT-OF-CONTACT IS 'CHIEF ROWE'

POC-PHONE IS '325-0880'

USER-ORGANIZATION IS ~~NIC-44~~



MEMBER NAME DIAS/SUB

SERVICE-HISTORY IS EXISTING-SERVICE

COMMENTS

- 'THE NIPSSA ADMINISTRATIVE REPORT CONTAINS ALL BILLETS
- 'WITH THEIR ASSOCIATED INCUMBENTS AND ACCESSES
- '(EXCLUDING 2 ACCESSES) FOR THE NIPSSA COMMAND
- 'CODES. NIPSSA BILLETS ARE DEFINED BY THE FOLLOWING
- 'SERIES OF COMMAND CODE/BILLET NUMBER RANGES:
- '223000/000 THROUGH 223000/999.
- '280232/000 THROUGH 280232/999. AND
- '280290/000 THROUGH 280290/999.
- 'THIS REPORT IS DELIVERED TO NIPSSA-01S.'.

ADD PROGRAM NAME IS S8152040

PREPARED BY 'ISI'

PROGRAM DESCRIPTION

'ACCESS MANAGEMENT REPORT'

ENTRY-SECURITY IS ENTRY-UNCLASSIFIED

DATA-SECURITY IS DATA-UNCLASSIFIED

OUTPUT-SECURITY IS OUT-CONFIDENTIAL

WITHIN SUBSYSTEM SECSS

OUTPUT-MODE IS REPORT-MODE

FREQUENCY IS ADHOC

SERVICE-PRIORITY IS NORMAL-PRIORITY

RESPONSE-TIME IS OVERNIGHT-RESPONSE

MODE IS BATCH

POINT-OF-CONTACT IS 'CHIEF ROVE'

POC-PHONE IS '325-0880'

USER-ORGANIZATION IS 'NIC-44'

SERVICE-HISTORY IS NEW-SERVICE

COMMENTS

- 'THE ACCESS MANAGEMENT REPORT IS A LISTING.
- 'OF BILLETS WITH INCUMBENTS, THAT CONTAINS ONLY
- 'THOSE BILLETS THAT HAVE BEEN INDICATED AS TEMPORARY
- '(FOR 'FLOATING' ACCESS MANAGEMENT). THE BILLETS TO
- 'BE SELECTED MUST ALSO BE WITHIN THE COMMAND CODE
- 'RANGE SUPPLIED BY THE USER. THE REPORT IS SORTED
- 'IN COMMAND CODE AND BILLET NUMBER SEQUENCE.'.

ADD PROGRAM NAME IS S8152041

PREPARED BY 'ISI'

PROGRAM DESCRIPTION

'SSO ADMINISTRATIVE 800000 REPORT'

ENTRY-SECURITY IS ENTRY-UNCLASSIFIED

DATA-SECURITY IS DATA-UNCLASSIFIED

OUTPUT-SECURITY IS OUT-CONFIDENTIAL

WITHIN SUBSYSTEM SECSS

OUTPUT-MODE IS REPORT-MODE

FREQUENCY IS ADHOC

SERVICE-PRIORITY IS NORMAL-PRIORITY

RESPONSE-TIME IS OVERNIGHT-RESPONSE

MODE IS BATCH

POINT-OF-CONTACT IS 'CHIEF ROVE'

POC-PHONE IS '325-0880'

USER-ORGANIZATION IS 'NIC-44'

SERVICE-HISTORY IS EXISTING-SERVICE

COMMENTS

- 'THE SSO ADMINISTRATIVE SERIES - 800XXX REPORT
- '(PREVIOUS JOB ID #37). THIS REPORT CONTAINS ONLY
- '800000 THROUGH 800999 COMMAND CODES WITH THEIR
- 'ASSOCIATED BILLETS AND INCUMBENTS. LISTED IN

MEMBER NAME DIASMSUB

- \* COMMAND CODE AND BILLET NUMBER SEQUENCE.\*

ADD PROGRAM NAME IS S8152042

PREPARED BY \*ISI\*

PROGRAM DESCRIPTION

\*SSO ADMIN -DUL- REPORT\*

ENTRY-SECURITY IS ENTRY-UNCLASSIFIED

DATA-SECURITY IS DATA-UNCLASSIFIED

OUTPUT-SECURITY IS OUT-CONFIDENTIAL

WITHIN SUBSYSTEM SECSS

OUTPUT-MODE IS REPORT-MODE

FREQUENCY IS ADHOC

SERVICE-PRIORITY IS NORMAL-PRIORITY

RESPONSE-TIME IS OVERNIGHT-RESPONSE

MODE IS BATCH

POINT-OF-CONTACT IS \*CHIEF ROWE\*

POC-PHONE IS \*325-0880\*

USER-ORGANIZATION IS \*NIC-44\*

SERVICE-HISTORY IS EXISTING-SERVICE

COMMENTS

- \*THE SSO ADMINISTRATIVE - DUL ACCESSES REPORT

- \* (PREVIOUS JOB ID 454) CONTAINS ALL BILLETS AND

- \* INCUMBENTS IN THE DATABASE WHO POSSESS DUL ACCESSES.

- \* LISTED IN COMMAND CODE/BILLET NUMBER SEQUENCE.\*

ADD PROGRAM NAME IS S8152043

PREPARED BY \*ISI\*

PROGRAM DESCRIPTION

\*OPNAV ADMINISTRATIVE REPORT\*

ENTRY-SECURITY IS ENTRY-UNCLASSIFIED

DATA-SECURITY IS DATA-UNCLASSIFIED

OUTPUT-SECURITY IS OUT-CONFIDENTIAL

WITHIN SUBSYSTEM SECSS

OUTPUT-MODE IS REPORT-MODE

FREQUENCY IS ADHOC

SERVICE-PRIORITY IS NORMAL-PRIORITY

RESPONSE-TIME IS OVERNIGHT-RESPONSE

MODE IS BATCH

POINT-OF-CONTACT IS \*CHIEF ROWE\*

POC-PHONE IS \*325-0880\*

USER-ORGANIZATION IS \*NIC-44\*

SERVICE-HISTORY IS EXISTING-SERVICE

COMMENTS

- \*THE OPNAV ADMINISTRATIVE REPORT

- \* (PREVIOUS JOB ID 458) CONTAINS ALL BILLETS

- \* WITH THEIR ASSOCIATED INCUMBENTS AND ACCESSES

- \* (EXCLUDING 2 ACCESSES) FOR THE COMMAND CODES

- \* SPECIFIED BELOW, LISTED IN SSO COMMAND, COMMAND

- \* CODE, AND BILLET NUMBER ORDER. COMMAND CODES USED

- \* FOR THIS REPORT ARE:

- \* 742000-742999,

- \* 772000-772999,

- \* 774000-774999,

- \* 782000-782999, AND

- \* 999900-999999.\*

ADD PROGRAM NAME IS S8152044

PREPARED BY \*ISI\*

PROGRAM DESCRIPTION

\*UPDATE OF ACCESS HISTORY RELATED DATA\*

ENTRY-SECURITY IS ENTRY-UNCLASSIFIED



MEMBER NAME DIASMSID

DATA-SECURITY IS DATA-UNCLASSIFIED  
CLIENT-SECURITY IS OUT-CONFIDENTIAL  
WITHIN SUBSYSTEM SECSS  
OUTPUT-MODE IS REPORT-MODE  
FREQUENCY IS DAILY  
SERVICE-PRIORITY IS NORMAL-PRIORITY  
RESPONSE-TIME IS OVERNIGHT-RESPONSE  
MODE IS BATCH  
POINT-OF-CONTACT IS 'CHIEF ROWE'  
POC-PHONE IS '325-0880'  
USER-ORGANIZATION IS 'NIC-44'  
SERVICE-HISTORY IS NEW-SERVICE  
COMMENTS

- 'THE ACCESS HISTORY DATA IN THE DATABASE
- 'MAY BE UPDATED TO INCLUDE ADDITIONS,
- 'CHANGE AND DELETIONS. A PRINTED
- 'TRANSACTION REPORT SHOWING ALL TRANSACTIONS
- 'MADE TO DATA BASE WILL BE PRODUCED'.

ADD PROGRAM NAME IS SM152045

PREPARED BY 'ISI'

PROGRAM DESCRIPTION

'RETRIEVAL OF ACCESS HISTORY RELATED DATA'

ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
DATA-SECURITY IS DATA-UNCLASSIFIED  
OUTPUT-SECURITY IS OUT-CONFIDENTIAL  
WITHIN ADHOC SECSS  
OUTPUT-MODE IS REPORT-MODE  
FREQUENCY IS ADHOC  
SERVICE-PRIORITY IS NORMAL-PRIORITY  
RESPONSE-TIME IS OVERNIGHT-RESPONSE  
MODE IS BATCH  
POINT-OF-CONTACT IS 'CHIEF ROWE'  
POC-PHONE IS '325-0880'  
USER-ORGANIZATION IS 'NIC-44'  
SERVICE-HISTORY IS NEW-SERVICE  
COMMENTS

- 'THIS REPORT CONTAINS ALL ACCESS HISTORY
- 'DATA THAT IS MAINTAINED IN THE DATABASE
- 'FOR THE SPECIFIC NAME AS SUPPLIED BY
- 'THE USER'.

ADD PROGRAM NAME IS SM152046

PREPARED BY 'ISI'

PROGRAM DESCRIPTION

'DYNAMIC UTILIZATION (FLOAT) OF ACCESS CODES'

ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
DATA-SECURITY IS DATA-UNCLASSIFIED  
OUTPUT-SECURITY IS OUT-CONFIDENTIAL  
WITHIN SUBSYSTEM SECSS  
FREQUENCY IS ADHOC  
SERVICE-PRIORITY IS NORMAL-PRIORITY  
RESPONSE-TIME IS OVERNIGHT-RESPONSE  
MODE IS BATCH  
POINT-OF-CONTACT IS 'CHIEF ROWE'  
POC-PHONE IS '325-0880'  
USER-ORGANIZATION IS 'NIC-44'  
SERVICE-HISTORY IS NEW-SERVICE  
COMMENTS

'THE ABILITY TO DYNAMICALLY ALLOCATE

MEMBER NAME DIASMSUN

- THE ACCESS CODES ('TICKETS') AMONG
- ALL ACTIVE BILLETS WILL BE PROVIDED
- AUTOMATICALLY FOR A SPECIFIC BILLET
- WHEN THE BILLET IS ADDED TO THE FILE.
- THIS CONTROL WILL PROVIDE FOR OPTIMUM
- USAGE OF THE ALLOWED NUMBER OF 'TICKETS'
- FOR WHICH THE NAVY HAS RESPONSIBILITY..

ADD PROGRAM NAME IS S4152047

PREPARED BY 'ISI'

PROGRAM DESCRIPTION

'ACCESS MANAGEMENT REPORT'

ENTRY-SECURITY IS ENTRY-UNCLASSIFIED

DATA-SECURITY IS DATA-UNCLASSIFIED

OUTPUT-SECURITY IS OUT-CONFIDENTIAL

WITHIN SUBSYSTEM SECSS

OUTPUT-MODE IS REPORT-MODE

FREQUENCY IS ADHOC

SERVICE-PRIORITY IS NORMAL-PRIORITY

RESPONSE-TIME IS OVERNIGHT-RESPONSE

MODE IS BATCH

POINT-OF-CONTACT IS 'CHIEF ROWE'

POC-PHONE IS '325-0880'

USER-ORGANIZATION IS 'NIC-44'

SERVICE-HISTORY IS NEW-SERVICE

COMMENTS

'THE ACCESS MANAGEMENT REPORT IS A

- LISTING OF BILLETS WITH INCUMMENTS THAT
- CONTAINS ONLY THOSE BILLETS THAT HAVE
- BEEN INDICATED AS TEMPORARY (FOR 'FLOATING'
- 'ACCESS MANAGEMENT'). THE BILLETS TO BE SELECTED
- MUST ALSO BE WITHIN THE COMMAND CODE RANGE
- SUPPLIED BY THE USER. THE REPORT IS SORTED
- IN COMMAND CODE AND BILLET NUMBER SEQUENCE..

ADD PROGRAM NAME IS S4152048

PREPARED BY 'ISI'

PROGRAM DESCRIPTION

'UPDATE OF FACILITIES ACCREDITATION DATA'

ENTRY-SECURITY IS ENTRY-UNCLASSIFIED

DATA-SECURITY IS DATA-UNCLASSIFIED

OUTPUT-SECURITY IS OUT-CONFIDENTIAL

WITHIN SUBSYSTEM SECSS

OUTPUT-MODE IS REPORT-MODE

FREQUENCY IS ADHOC

SERVICE-PRIORITY IS NORMAL-PRIORITY

RESPONSE-TIME IS OVERNIGHT-RESPONSE

MODE IS BATCH

POINT-OF-CONTACT IS 'CARL SOWELL'

POC-PHONE IS '325-0317'

USER-ORGANIZATION IS 'NIC-44'

SERVICE-HISTORY IS EXISTING-SERVICE

COMMENTS

'THE FACILITIES ACCREDITATION DATA IN

- THE DATABASE MAYBE UPDATED TO INCLUDE
- ADDITIONS, CHANGE AND DELETIONS. A
- PRINTED TRANSACTION REPORT SHOWING ALL
- TRANSACTIONS MADE TO THE DATABASE WILL
- BE PRODUCED..

ADD PROGRAM NAME IS S4152049

MEMBER NAME DIASMSUB

PREPARED BY 'ISI'

PROGRAM DESCRIPTION

'LIST OF SPACES WITHOUT FINAL ACCREDITATION'

ENTRY-SECURITY IS ENTRY-UNCLASSIFIED

DATA-SECURITY IS DATA-UNCLASSIFIED

OUTPUT-SECURITY IS OUT-CONFIDENTIAL

WITHIN SUBSYSTEM SECSS

OUTPUT-MODE IS REPORT-MODE

FREQUENCY IS ADHOC

SERVICE-PRIORITY IS NORMAL-PRIORITY

RESPONSE-TIME IS OVERNIGHT-RESPONSE

MODE IS BATCH

POINT-OF-CONTACT IS 'CARL SOWELL'

POC-PHONE IS '325-4317'

USER-ORGANIZATION IS 'NIC-44'

SERVICE-HISTORY IS EXISTING-SERVICE

COMMENTS

'THIS REPORT WILL PRODUCE A LIST OF ALL

- 'RECORDS IN WHICH THE FINAL ACCREDITATION

- 'IS BLANK. THE REPORT IS USED BY NIC-44 TO

- 'DETERMINE WHY SPACES HAVE NOT RECEIVED

- 'THEIR FINAL ACCREDITATION.'

ADD PROGRAM NAME IS S8152050

PREPARED BY 'ISI'

PROGRAM DESCRIPTION

'FACILITIES ACCREDITATION LIST OF SPACE'

ENTRY-SECURITY IS ENTRY-UNCLASSIFIED

DATA-SECURITY IS DATA-UNCLASSIFIED

OUTPUT-SECURITY IS OUT-CONFIDENTIAL

WITHIN SUBSYSTEM SECSS

OUTPUT-MODE IS REPORT-MODE

FREQUENCY IS ADHOC

SERVICE-PRIORITY IS NORMAL-PRIORITY

RESPONSE-TIME IS OVERNIGHT-RESPONSE

MODE IS BATCH

POINT-OF-CONTACT IS 'CARL SOWELL'

POC-PHONE IS '325-0317'

USER-ORGANIZATION IS 'NIC-44'

SERVICE-HISTORY IS EXISTING-SERVICE

COMMENTS

'THIS REPORT WILL PRODUCE A LIST OF

- 'ALL ROOMS THAT ARE ACCREDITED. THE REPORT

- 'IS IN COMMAND CODE/BUILDING NUMBER/ROOM

- 'NUMBER SEQUENCE.'

ADD PROGRAM NAME IS S8152051

PREPARED BY 'ISI'

PROGRAM DESCRIPTION

'FACILITIES ACCREDITATION REINSPECTION REPORT'

ENTRY-SECURITY IS ENTRY-UNCLASSIFIED

DATA-SECURITY IS DATA-UNCLASSIFIED

OUTPUT-SECURITY IS OUT-CONFIDENTIAL

WITHIN SUBSYSTEM SECSS

OUTPUT-MODE IS REPORT-MODE

FREQUENCY IS ADHOC

SERVICE-PRIORITY IS NORMAL-PRIORITY

RESPONSE-TIME IS OVERNIGHT-RESPONSE

MODE IS BATCH

POINT-OF-CONTACT IS 'CARL SOWELL'

MEMBER NAME DIASMSUP

POC-PHONE IS '325-0317'

USER-ORGANIZATION IS 'NIC-44'

SERVICE-HISTORY IS EXISTING-SERVICE

COMMENTS

- 'THIS REPORT LISTS ALL SPACES THAT ARE
- 'DUE FOR REINSPECTION IN THE NEXT THREE MONTHS.
- 'A DATE, ENTERED BY THE USER, WHICH SPECIFIES
- 'THE STARTING DATE IS ENTERED INTO THE
- 'SYSTEM. THE REPORT IS IN COMMAND CODE/BUILDING
- 'NUMBER/ROOM NUMBER SEQUENCE..'

ADD PROGRAM NAME IS S8152052

PREPARED BY 'ISI'

PROGRAM DESCRIPTION

'FACILITIES ACCREDITATION SEMI-ANNUAL REPORT'

ENTRY-SECURITY IS ENTRY-UNCLASSIFIED

DATA-SECURITY IS DATA-UNCLASSIFIED

OUTPUT-SECURITY IS OUT-CONFIDENTIAL

WITHIN SUBSYSTEM SECSS

OUTPUT-MODE IS REPORT-MODE

FREQUENCY IS SEMIANNUAL

SERVICE-PRIORITY IS NORMAL-PRIORITY

RESPONSE-TIME IS OVERNIGHT-RESPONSE

MODE IS BATCH

POINT-OF-CONTACT IS 'CARL SOWELL'

POC-PHONE IS '325-0317'

USER-ORGANIZATION IS 'NIC-44'

SERVICE-HISTORY IS EXISTING-SERVICE

COMMENTS

- 'THIS REPORT IS A LIST OF ALL SPACES TO BE
- 'USED BY PARENT ACTIVITIES TO KEEP A RECORD
- 'OF THEIR SPACES. THE REPORT IS IN COMMAND
- 'CODE/BUILDING NUMBER/ROOM NUMBER SEQUENCE
- 'WITH PAGE BREAKS AT COMMAND CODE CHANGES.
- 'THE COMMANDS TO BE PRINTED ARE ENTERED BY THE
- 'USER AS A LOW TO HIGH COMMAND CODE NUMBER
- 'RANGE..'

ADD PROGRAM NAME IS S8152053

PREPARED BY 'ISI'

PROGRAM DESCRIPTION

'FACILITY LIST REPORT'

ENTRY-SECURITY IS ENTRY-UNCLASSIFIED

DATA-SECURITY IS DATA-UNCLASSIFIED

OUTPUT-SECURITY IS OUT-CONFIDENTIAL

WITHIN SUBSYSTEM SECSS

OUTPUT-MODE IS REPORT-MODE

FREQUENCY IS ADHOC

SERVICE-PRIORITY IS NORMAL-PRIORITY

RESPONSE-TIME IS OVERNIGHT-RESPONSE

MODE IS BATCH

POINT-OF-CONTACT IS 'CARL SOWELL'

POC-PHONE IS '325-0317'

USER-ORGANIZATION IS 'NIC-44'

SERVICE-HISTORY IS EXISTING-SERVICE

COMMENTS

- 'THIS REPORT PRODUCES A LIST OF ALL SPACES
- 'HAVING 00 IN THE LAST TWO CHARACTERS OF
- 'THE CONTROL NUMBER. THIS REPORT IS ORDERED
- 'BY CONTROL NUMBER AND IS PRODUCED UPON



MEMBER NAME DIASMSUB

- REQUEST OF THE USER...  
ADD PROGRAM NAME IS S8152054

PREPARED BY 'ISI'

PROGRAM DESCRIPTION

'TSCM REQUIRED SPACES'

ENTRY-SECURITY IS ENTRY-UNCLASSIFIED

DATA-SECURITY IS DATA-UNCLASSIFIED

OUTPUT-SECURITY IS OUT-CONFIDENTIAL

WITHIN SUBSYSTEM SECSS

OUTPUT-MODE IS REPORT-MODE

FREQUENCY IS ADHOC

SERVICE-PRIORITY IS NORMAL-PRIORITY

RESPONSE-TIME IS OVERNIGHT-RESPONSE

MODE IS BATCH

POINT-OF-CONTACT IS 'CARL SOWELL'

POC-PHONE IS '325-0317'

USER-ORGANIZATION IS 'NIC-44'

SERVICE-HISTORY IS EXISTING-SERVICE

COMMENTS

'THIS REPORT PRODUCES A LIST OF ALL

'SPACES REQUIRING TECHNICAL SURVEILLANCE

'COUNTER MEASURES (TSCM) AND IS PRODUCED

- UPON USER REQUEST...  
ADD PROGRAM NAME IS S8152055

PREPARED BY 'ISI'

PROGRAM DESCRIPTION

'UPDATE OF RILLET RELATED DATA'

ENTRY-SECURITY IS ENTRY-UNCLASSIFIED

DATA-SECURITY IS DATA-UNCLASSIFIED

OUTPUT-SECURITY IS OUT-CONFIDENTIAL

WITHIN SUBSYSTEM SECSS

OUTPUT-MODE IS DISPLAY-MODE

FREQUENCY IS DAILY

SERVICE-PRIORITY IS NORMAL-PRIORITY

RESPONSE-TIME IS IMMEDIATE-RESPONSE

MODE IS SHADOW

POINT-OF-CONTACT IS 'CHIEF ROWE'

POC-PHONE IS '325-0880'

USER-ORGANIZATION IS 'NIC-44'

SERVICE-HISTORY IS NEW-SERVICE

COMMENTS

'THE RILLET RELATED DATA IN THE DATABASE

'MAY BE UPDATED TO INCLUDE ADDITIONS, CHANGES

'AND DELETIONS, AND TO ALSO DEFINE RELATIONSHIPS

'WITH INCUMBENTS. A HARD COPY OF THE TRANSACTION

'MAY BE PRODUCED AT THE OPTION OF THE USER.

'THE INPUT OF DATA CAN BE SEVERAL SEQUENTIALLY

'ACCESSED FORMATTED SCREENS. THE SCREEN IS DEFINED

'TO BE 24 LINES OF 80 CHARACTERS ON THE DISPLAY

'TERMINAL...  
ADD PROGRAM NAME IS S8152056

PREPARED BY 'ISI'

PROGRAM DESCRIPTION

'UPDATE OF INCUMBENT RELATED DATA'

ENTRY-SECURITY IS ENTRY-UNCLASSIFIED

DATA-SECURITY IS DATA-UNCLASSIFIED

OUTPUT-SECURITY IS OUT-CONFIDENTIAL

WITHIN SUBSYSTEM SECSS



MEMBER NAME DIASMSUB

OUTPUT-MODE IS DISPLAY-MODE

FREQUENCY IS DAILY

SERVICE-PRIORITY IS NORMAL-PRIORITY

RESPONSE-TIME IS IMMEDIATE-RESPONSE

MODE IS SHADOW

POINT-OF-CONTACT IS 'CHIEF ROWE'

POC-PHONE IS '325-0880'

USER-ORGANIZATION IS 'NIC-44'

SERVICE-HISTORY IS NEW-SERVICE

COMMENTS

- 'THE INCUMBENT RELATED DATA IN THE DATABASE
- 'MAY BE UPDATED TO INCLUDE ADDITIONS, CHANGES
- 'AND DELETIONS, AND TO ALSO DEFINE RELATIONSHIPS
- 'WITH BILLETS. A HARD COPY OF THE TRANSACTION
- 'MAY BE PRODUCED AT THE OPTION OF THE USER. THE
- 'INPUT OF DATA CAN BE SEVERAL SEQUENTIALLY
- 'ACCESSED FORMATTED SCREENS..'

ADD PROGRAM NAME IS S8152057

PREPARED BY 'ISI'

PROGRAM DESCRIPTION

'RETRIEVAL OF USER DEFINED DATA'

ENTRY-SECURITY IS ENTRY-UNCLASSIFIED

DATA-SECURITY IS DATA-UNCLASSIFIED

OUTPUT-SECURITY IS OUT-CONFIDENTIAL

WITHIN SUBSYSTEM SECSS

OUTPUT-MODE IS DISPLAY-MODE

FREQUENCY IS ADHOC

SERVICE-PRIORITY IS NORMAL-PRIORITY

RESPONSE-TIME IS IMMEDIATE-RESPONSE

MODE IS SHADOW

POINT-OF-CONTACT IS 'CHIEF ROWE'

POC-PHONE IS '325-0880'

USER-ORGANIZATION IS 'NIC-44'

SERVICE-HISTORY IS NEW-SERVICE

COMMENTS

- 'ANY DATA IN THE SECURITY MANAGEMENT
- 'SUBSYSTEM DATABASE MAY BE DISPLAYED TO
- 'THE USER BY A USER-SPECIFIED QUERY.
- 'THIS QUERY DEFINES TO THE SYSTEM THE SPECIFIC
- 'DATA THAT THE USER REQUIRES. A HARD COPY OF
- 'THE INFORMATION MAY BE PRODUCED AT THE OPTION
- 'OF THE USER. THE OUTPUT OF DATA CAN BE
- 'DISPLAYED ON SEVERAL SEQUENTIALLY ACCESSED
- 'FORMATTED SCREENS..'

ADD PROGRAM NAME IS S8152058

PREPARED BY 'ISI'

PROGRAM DESCRIPTION

'SELECTIVE BILLET REPORT'

ENTRY-SECURITY IS ENTRY-UNCLASSIFIED

DATA-SECURITY IS DATA-UNCLASSIFIED

OUTPUT-SECURITY IS OUT-CONFIDENTIAL

WITHIN SUBSYSTEM SECSS

OUTPUT-MODE IS DISPLAY-MODE

FREQUENCY IS ADHOC

SERVICE-PRIORITY IS NORMAL-PRIORITY

RESPONSE-TIME IS RESPONSE-1

MODE IS SHADOW

POINT-OF-CONTACT IS 'CHIEF ROWE'

MEMBER NAME DIASMSUB  
 FOC-PHONE IS '325-0880'  
 USER-ORGANIZATION IS 'NIC-44'  
 SERVICE-HISTORY IS NEW-SERVICE  
 COMMENTS  
 - 'THIS REPORT CONTAINS ALL BILLETS AND  
 - 'INCUMBENTS IN THE DATABASE WITHIN THE  
 - 'USER DEFINED FROM-AND-THROUGH PARAMETER.  
 - 'THE FROM-AND-THROUGH PARAMETER FOR THIS  
 - 'REQUEST CONSISTS OF THE LOW AND THE HIGH  
 - 'COMMAND-BILLET NUMBERS THAT DETERMINE  
 - 'THE RANGE TO BE SELECTED. IT IS SORTED IN  
 - 'COMMAND CODE AND BILLET NUMBER SEQUENCE.  
 - 'DATA FOR THIS REPORT INCLUDES BOTH  
 - 'INCUMBENT, BILLET, AND ACCESS DATA.'  
 ADD PROGRAM NAME IS SMIS2059  
 PREPARED BY 'ISI'  
 PROGRAM DESCRIPTION  
 'SELECTIVE BILLET REPORT BY INCUMBENT'  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-UNCLASSIFIED  
 OUTPUT-SECURITY IS OUT-CONFIDENTIAL  
 WITHIN-SUBSYSTEM SECSS  
 OUTPUT-MODE IS DISPLAY-MODE  
 FREQUENCY IS ADHOC  
 SERVICE-PRIORITY IS NORMAL-PRIORITY  
 RESPONSE-TIME IS RESPONSE-1  
 MODE IS SHADOW  
 POINT-OF-CONTACT IS 'CHIEF ROWE'  
 FOC-PHONE IS '325-0880'  
 USER-ORGANIZATION IS 'NIC-44'  
 SERVICE-HISTORY IS NEW-SERVICE  
 COMMENTS  
 - 'THIS REPORT CONTAINS ALL BILLETS AND  
 - 'INCUMBENTS IN THE DATABASE WITHIN THE  
 - 'USER DEFINED FROM-AND-THROUGH PARAMETER.  
 - 'THE FROM-AND-THROUGH PARAMETER FOR THIS  
 - 'REQUEST CONSISTS OF THE LOW AND THE HIGH  
 - 'INCUMBENT NAMES THAT DETERMINE THE  
 - 'ALPHABETIC RANGE TO BE SELECTED. IT IS  
 - 'SORTED ALPHABETICALLY BY INCUMBENT NAME.  
 - 'DATA FOR THIS REPORT INCLUDES BOTH INCUMBENT,  
 - 'BILLET, AND ACCESS DATA.'  
 ADD PROGRAM NAME IS SMIS2060  
 PREPARED BY 'ISI'  
 PROGRAM DESCRIPTION  
 'SELECTIVE BILLET REPORT BY CS#1'  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-UNCLASSIFIED  
 OUTPUT-SECURITY IS OUT-CONFIDENTIAL  
 WITHIN-SUBSYSTEM SECSS  
 OUTPUT-MODE IS DISPLAY-MODE  
 FREQUENCY IS ADHOC  
 SERVICE-PRIORITY IS NORMAL-PRIORITY  
 RESPONSE-TIME IS RESPONSE-1  
 MODE IS SHADOW  
 POINT-OF-CONTACT IS 'CHIEF ROWE'  
 FOC-PHONE IS '325-0880'  
 USER-ORGANIZATION IS 'NIC-44'

MEMBER NAME DIASMSUB

SERVICE-HISTORY IS NEW-SERVICE

COMMENTS

- 'THIS REPORT CONTAINS ALL BILLETS AND INCUMBENTS
- 'IN THE DATABASE WITHIN THE USER DEFINED
- 'FROM-AND-THROUGH PARAMETER. THE FROM-AND-THROUGH
- 'PARAMETER FOR THIS REQUEST CONSISTS OF THE LOW AND
- 'THE HIGH SOCIAL SECURITY NUMBERS THAT DETERMINE
- 'THE RANGE TO BE SELECTED. IT IS SORTED NUMERICALLY
- 'BY THE SOCIAL SECURITY NUMBER. DATA FOR THIS REPORT
- 'INCLUDES BOTH INCUMBENT, BILLET, AND ACCESS DATA.'.

ADD PROGRAM NAME IS SB152061

PREPARED BY 'ISI'

PROGRAM DESCRIPTION

'ACCESS CODE SUMMARY REPORT'

ENTRY-SECURITY IS ENTRY-UNCLASSIFIED

DATA-SECURITY IS DATA-UNCLASSIFIED

OUTPUT-SECURITY IS OUT-CONFIDENTIAL

WITHIN SUBSYSTEM SECSS

OUTPUT-MODE IS DISPLAY-MODE

FREQUENCY IS ADHOC

SERVICE-PRIORITY IS NORMAL-PRIORITY

RESPONSE-TIME IS RESPONSE-1

MODE IS SHADOW

POINT-OF-CONTACT IS 'CHIEF ROWE'

POC-PHONE IS '325-0880'

USER-ORGANIZATION IS 'NIC-44'

SERVICE-HISTORY IS NEW-SERVICE

COMMENTS

- 'THIS REPORT CONTAINS A SUMMARY OF THE
- 'NUMBER OF ACCESSES IN USE FOR ALL TYPES
- 'OF ACCESSES. IT IS LISTED ALPHABETICALLY
- 'BY ACCESS CODE. DATA FOR THIS REPORT USES
- 'BILLET AND AND ACCESS CODE DATA.'.

ADD PROGRAM NAME IS SB152062

PREPARED BY 'ISI'

PROGRAM DESCRIPTION

'FLOATING ACCESS CODE SUMMARY REPORT'

ENTRY-SECURITY IS ENTRY-UNCLASSIFIED

DATA-SECURITY IS DATA-UNCLASSIFIED

OUTPUT-SECURITY IS OUT-CONFIDENTIAL

WITHIN SUBSYSTEM SECSS

OUTPUT-MODE IS DISPLAY-MODE

FREQUENCY IS ADHOC

SERVICE-PRIORITY IS NORMAL-PRIORITY

RESPONSE-TIME IS RESPONSE-1

MODE IS SHADOW

POINT-OF-CONTACT IS 'CHIEF ROWE'

POC-PHONE IS '325-0880'

USER-ORGANIZATION IS 'NIC-44'

SERVICE-HISTORY IS NEW-SERVICE

COMMENTS

- 'THIS REPORT CONTAINS A DETAILED STATUS
- 'OF FLOATING CONTROL INFORMATION FOR ALL
- 'TYPES OF ACCESSES. IT IS LISTED ALPHABETICALLY
- 'BY ACCESS CODE. DATA FOR THIS REPORT USES
- 'BILLET AND ACCESS CODE DATA.'.

ADD PROGRAM NAME IS SB152063

PREPARED BY 'ISI'

MEMBER NAME DIASMSUB

PROGRAM DESCRIPTION

BRANCH OF SERVICE SUMMARY REPORT

ENTRY-SECURITY IS ENTRY-UNCLASSIFIED

DATA-SECURITY IS DATA-UNCLASSIFIED

OUTPUT-SECURITY IS OUT-CONFIDENTIAL

WITHIN SUBSYSTEM SECSS

OUTPUT-MODE IS DISPLAY-MODE

FREQUENCY IS ADHOC

SERVICE-PRIORITY IS NORMAL-PRIORITY

RESPONSE-TIME IS RESPONSE-1

MODE IS SHADOW

POINT-OF-CONTACT IS 'CHIEF ROWE'

POC-PHONE IS '325-0880'

USER-ORGANIZATION IS 'NIC-44'

SERVICE-HISTORY IS NEW-SERVICE

COMMENTS

'THIS REPORT CONTAINS A SUMMARY OF THE NUMBER

'OF ACCESSES IN USE FOR ALL TYPES OF ACCESS

'BY BRANCH OF SERVICE IN THE DATABASE. IT IS

'LISTED ALPHABETICALLY BY ACCESS CODE. DATA

'FOR THIS REPORT USES INCUMBENT AND ACCESS DATA.'.

ADD PROGRAM-NAME IS S8152064

PREPARED BY 'ISI'

PROGRAM DESCRIPTION

UPDATE OF ACCESS HISTORY RELATED DATA

ENTRY-SECURITY IS ENTRY-UNCLASSIFIED

DATA-SECURITY IS DATA-UNCLASSIFIED

OUTPUT-SECURITY IS OUT-CONFIDENTIAL

WITHIN SUBSYSTEM SECSS

OUTPUT-MODE IS DISPLAY-MODE

FREQUENCY IS ADHOC

SERVICE-PRIORITY IS NORMAL-PRIORITY

RESPONSE-TIME IS IMMEDIATE-RESPONSE

MODE IS SHADOW

POINT-OF-CONTACT IS 'CHIEF ROWE'

POC-PHONE IS '325-0880'

USER-ORGANIZATION IS 'NIC-44'

SERVICE-HISTORY IS NEW-SERVICE

COMMENTS

'THE ACCESS HISTORY RELATED DATA IN THE

'DATABASE MAY BE UPDATED TO INCLUDE

'ADDITIONS, CHANGES AND DELETIONS. A

'HARD COPY OF THE TRANSACTION MAY BE

'PRODUCED AT THE OPTION OF THE USER.

'THE INPUT OF DATA CAN BE SEVERAL SEQUENTIALLY

'ACCESSED FORMATTED SCREENS.'.

ADD PROGRAM NAME IS S8152065

PREPARED BY 'ISI'

PROGRAM DESCRIPTION

SELECTIVE ACCESS HISTORY REPORT BY INCUMBENT

ENTRY-SECURITY IS ENTRY-UNCLASSIFIED

DATA-SECURITY IS DATA-UNCLASSIFIED

OUTPUT-SECURITY IS OUT-CONFIDENTIAL

WITHIN SUBSYSTEM SECSS

OUTPUT-MODE IS DISPLAY-MODE

FREQUENCY IS ADHOC

SERVICE-PRIORITY IS NORMAL-PRIORITY

RESPONSE-TIME IS RESPONSE-1



MEMBER NAME DIASMSUB  
MODE IS SHADOW  
POINT-OF-CONTACT IS 'CHIEF ROWE'  
POC-PHONE IS '325-0880'  
USER-ORGANIZATION IS 'NIC-44'  
SERVICE-HISTORY IS NEW-SERVICE  
COMMENTS  
- 'THIS REPORT CONTAINS ALL ACCESS HISTORY  
- 'DATA THAT IS MAINTAINED IN THE DATABASE  
- 'FOR THE SPECIFIC NAME AS SUPPLIED BY  
- 'THE USER. THE REQUESTER CAN VIEW THE  
- 'REPORT AT THE USER'S TERMINAL AFTER THE  
- 'QUERY IS ENTERED.'  
ADD PROGRAM NAME IS S0152066  
PREPARED BY 'ISI'  
PROGRAM DESCRIPTION  
'DYNAMIC UTILIZATION (FLOAT) OF ACCESS CODES'  
ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
DATA-SECURITY IS DATA-UNCLASSIFIED  
OUTPUT-SECURITY IS OUT-CONFIDENTIAL  
WITHIN SUBSYSTEM SECSS  
OUTPUT-MODE IS DISPLAY-MODE  
FREQUENCY IS ADHOC  
SERVICE-PRIORITY IS NORMAL-PRIORITY  
RESPONSE-TIME IS IMMEDIATE-RESPONSE  
MODE IS SHADOW  
POINT-OF-CONTACT IS 'CHIEF ROWE'  
POC-PHONE IS '325-0880'  
USER-ORGANIZATION IS 'NIC-44'  
SERVICE-HISTORY IS NEW-SERVICE  
COMMENTS  
- 'THE ABILITY TO DYNAMICALLY ALLOCATE THE  
- 'ACCESS CODES ('TICKETS') AMONG ALL ACTIVE  
- 'BILLETS WILL BE PROVIDED AUTOMATICALLY FOR  
- 'A SPECIFIC BILLET WHEN THE BILLET IS ADDED  
- 'TO THE FILE. THIS CONTROL WILL PROVIDE FOR  
- 'OPTIMUM USAGE OF THE ALLOWED NUMBER OF 'TICKETS'  
- 'FOR WHICH THE NAVY HAS RESPONSIBILITY.'  
ADD PROGRAM NAME IS S0152067  
PREPARED BY 'ISI'  
PROGRAM DESCRIPTION  
'UPDATE OF FACILITIES ACCREDITATION DATA'  
ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
DATA-SECURITY IS DATA-UNCLASSIFIED  
OUTPUT-SECURITY IS OUT-CONFIDENTIAL  
WITHIN SUBSYSTEM SECSS  
OUTPUT-MODE IS DISPLAY-MODE  
FREQUENCY IS ADHOC  
SERVICE-PRIORITY IS NORMAL-PRIORITY  
RESPONSE-TIME IS IMMEDIATE-RESPONSE  
MODE IS SHADOW  
POINT-OF-CONTACT IS 'CARL SOWELL'  
POC-PHONE IS '325-0317'  
USER-ORGANIZATION IS 'NIC-44'  
SERVICE-HISTORY IS NEW-SERVICE  
COMMENTS  
- 'THE FACILITIES ACCREDITATION RELATED DATA  
- 'IN THE DATABASE MAY BE UPDATED TO INCLUDE  
- 'ADDITIONS, CHANGES AND DELETIONS. A HARD



MEMBER NAME DIASMSUB

- \*COPY OF THE TRANSACTION MAY BE PRODUCED AT
- \*THE OPTION OF THE USER. THE INPUT OF DATA
- \*CAN BE SEVERAL SEQUENTIALLY ACCESSED
- \*FORMATTED SCREENS.\*.

MEMBER NAME DIESMS  
 ADD ELEMENT NAME IS ACCESS-CCCE  
 PREPARED BY 'SOC-ISI'  
 ELEMENT DESCRIPTION IS  
 'ACCESS (TICKET) CODES.'  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-CONFIDENTIAL  
 APPLICATION-SYSTEM IS SECS  
 SERVICE-SUPPORTED IS SB152CC6  
 PICTURE IS XXXX  
 USAGE IS DISPLAY  
 VALUE IS SPACES  
 STORE-VALIDATION IS SSZA  
 MODIFY-VALIDATION IS SMZE  
 ELEMENT DESIGNATOR IS KEY  
 ELEMENT DESIGNATOR IS NC-VERIFY  
 ELEMENT DESIGNATOR IS VARIABLE  
 PRESENCE IS PRESENCE-REQUIRED  
 ELEMENT DEFINITION IS  
 - 'THIS FIELD CONTAINS THE TYPE OF SPECIAL INTELLIGENCE  
 - 'ACCESSES THAT ARE AVAILABLE TO NAVAL INTELLIGENCE COMMAND  
 'FOR ASSIGNING TO A BILLET OF A PERSON.'  
 SOURCE-DOC IS NONE  
 DATA-SUBJECT IS 'SECURITY, GENERAL'  
 USER IS 'NIC-44'  
 RESPONSIBLE FOR DEFINITION  
 COMMENTS  
 'THE ENTRY MAY BE 2, 3 OR 4 CHARACTERS IN  
 'LENGTH. VALIDATION IS FOR CURRENCY.'  
 ADD ELEMENT NAME IS BADG-DISPOSITION  
 PREPARED BY 'SOC-ISI'  
 ELEMENT DESCRIPTION IS  
 'DISPOSITION OF THE BADGE'  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-UNCLASSIFIED  
 APPLICATION-SYSTEM IS SECS  
 SERVICE-SUPPORTED IS SB152CC1  
 PICTURE IS X  
 USAGE IS DISPLAY  
 VALUE IS SPACES  
 STORE-VALIDATION IS SSZI  
 MODIFY-VALIDATION IS SAZI  
 ELEMENT DESIGNATOR IS FIXED  
 PRESENCE IS PRESENCE-OPTIONAL  
 ELEMENT DEFINITION IS  
 - 'THIS FIELD INDICATES THE DISPOSITION OF A  
 - 'BADGE. WHETHER IT IS STILL IN USE, DESTROYED OR LOST.'  
 SOURCE-DOC IS NONE  
 DATA-SUBJECT IS 'SECURITY, GENERAL'  
 USER IS 'NIC-44'  
 RESPONSIBLE FOR DEFINITION  
 RANGE IS 'U'  
 RANGE IS 'D'  
 RANGE IS 'L'  
 COMMENTS  
 'THE FIELD MUST CONTAIN ONE OF THE FOLLOWING VALUES  
 - ' U IN USE  
 - ' D DESTROYED  
 - ' L LOST.

MEMBER NAME DIAGMS

- ADD \*A NEW BADGE IS ISSUED EVERY 2 YEARS.\*

ELEMENT NAME IS BADGE-NUMBER

PREPARED BY \*SOC-ISI\*

ELEMENT DESCRIPTION IS

\*ASSIGNED BADGE NUMBER\*

ENTRY-SECURITY IS ENTRY-UNCLASSIFIED

DATA-SECURITY IS DATA-UNCLASSIFIED

APPLICATION-SYSTEM IS SECSS

SERVICE-SUPPORTED IS SB152CC1

PICTURE IS XXXXX

USAGE IS DISPLAY

VALUE IS SPACES

STORE-VALIDATION IS SSZA

MODIFY-VALIDATION IS SMZB

ELEMENT DESIGNATOR IS KEY

ELEMENT DESIGNATOR IS VARIABLE

PRESNCE IS PRESNCE-REQUIRED

ELEMENT DEFINITION IS

\*THE BADGE NUMBER IS A SEQUENTIAL ALPHA-NUMERIC

\*NUMBER ASSIGNED WITHIN A COMMAND.\*

SOURCE-DOC IS NONE

DATA-SUBJECT IS \*SECURITY, GENERAL\*

USER IS \*NIC-44\*

RESPONSIBLE FOR DEFINITION

COMMENTS

\*THE BADGE NUMBER WILL ALWAYS BE A LETTER FOLLOWED

\*BY A FOUR DIGIT NUMBER.\*

- ADD ELEMENT NAME IS BADGE-TYPE

PREPARED BY \*SOC-ISI\*

ELEMENT DESCRIPTION IS

\*BADGE TYPE\*

ENTRY-SECURITY IS ENTRY-UNCLASSIFIED

DATA-SECURITY IS DATA-UNCLASSIFIED

APPLICATION-SYSTEM IS SECSS

SERVICE-SUPPORTED IS SB152CC1

PICTURE IS X

USAGE IS DISPLAY

VALUE IS SPACES

STORE-VALIDATION IS SSZI

MODIFY-VALIDATION IS SMZI

ELEMENT DESIGNATOR IS KEY

ELEMENT DESIGNATOR IS VARIABLE

PRESNCE IS PRESNCE-REQUIRED

ELEMENT DEFINITION IS

\*THIS FIELD CONTAINS THE CODE INDICATING THE

\*TYPE OF BADGE.\*

SOURCE-DOC IS NONE

DATA-SUBJECT IS \*SECURITY, GENERAL\*

USER IS \*NIC-44\*

RESPONSIBLE FOR DEFINITION

RANGE IS \*K\*

RANGE IS \*C\*

RANGE IS \*L\*

RANGE IS \*U\*

RANGE IS \*C\*

RANGE IS \*U\*

RANGE IS \*U\*

COMMENTS

\*THE ENTRY IS A CODE FROM THE FOLLOWING TABLE

MEMBER NAME DIESMS

- R FOR A REGULAR (STANDARD) BADGE.
- C FOR A COURIER BADGE.
- L FOR A LTASON BADGE.
- U FOR A RESERVE BADGE.
- E FOR A ESCORTED VISITOR BADGE.
- U FOR A UNESCORTED VISITOR BADGE..

ADD ELEMENT NAME IS BI-CASE-CONTROL  
 PREPARED BY 'SOC-ISI'  
 ELEMENT DESCRIPTION IS  
 'BI CASE CONTROL NUMBER'  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-UNCLASSIFIED  
 APPLICATION-SYSTEM IS SECSS  
 SERVICE-SUPPORTED IS SB152CC4  
 PICTURE IS XT(9)  
 USAGE IS DISPLAY  
 VALUE IS SPACES  
 STORE-VALIDATION IS SSZG  
 MODIFY-VALIDATION IS SMZA  
 ELEMENT DESIGNATOR IS VARIABLE  
 PRESENCE IS PRESENCE-OPTIONAL  
 ELEMENT DEFINITION IS  
 'THIS FIELD CONTAINS THE CASE CONTROL NUMBER OF A  
 - BACKGROUND INVESTIGATION FOR A PERSON.'  
 SOURCE-DCC IS NONE  
 DATA-SUBJECT IS 'SECURITY, GENERAL'  
 USER IS 'NIC-44'  
 RESPONSIBLE FOR DEFINITION.

ADD ELEMENT NAME IS BI-PENDING-IND  
 PREPARED BY 'SOC-ISI'  
 ELEMENT DESCRIPTION IS  
 'BACKGROUND INVESTIGATION PENDING INDICATOR.'  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-UNCLASSIFIED  
 APPLICATION-SYSTEM IS SECSS  
 SERVICE-SUPPORTED IS SB152CC4  
 PICTURE IS X  
 USAGE IS DISPLAY  
 VALUE IS SPACES  
 STORE-VALIDATION IS SSZJ  
 MODIFY-VALIDATION IS SMZH  
 ELEMENT DESIGNATOR IS FIXED  
 PRESENCE IS PRESENCE-OPTIONAL  
 ELEMENT DEFINITION IS  
 'THIS FIELD IS USED TO INDICATE WHETHER OR NOT A BACKGROUND  
 - INVESTIGATION IS COMPLETE OR PENDING.'  
 SOURCE-DCC IS NONE  
 DATA-SUBJECT IS 'SECURITY, GENERAL'  
 USER IS 'NIC-44'  
 RESPONSIBLE FOR DEFINITION

RANGE IS 'C'  
 RANGE IS 'P'  
 RANGE IS 'I'  
 COMMENTS  
 'THE BACKGROUND INVESTIGATION PENDING INDICATOR IS SET TO  
 - OF THREE VALUES  
 - A 'C' MEANING A BACKGROUND INVESTIGATION IS COMPLETE,  
 - A 'P' MEANING A BACKGROUND INVESTIGATION IS PENDING, OR



MEMBER NAME DIESMS

- 'A' (SPACE) MEANING THAT ONE IS NOT STARTED..

ADD ELEMENT NAME IS BILLET-NUMBER

PREPARED BY 'SDC-ISI'

ELEMENT DESCRIPTION IS

'NUMBER ASSIGNED TO THE BILLET'

ENTRY-SECURITY IS ENTRY-UNCLASSIFIED

DATA-SECURITY IS DATA-UNCLASSIFIED

APPLICATION-SYSTEM IS SECSS

SERVICE-SUPPORTED IS S8152CC6

PICTURE IS 9(4)

USAGE IS DISPLAY

VALUE IS ZEROS

STORE-VALIDATION IS SSZM

MODIFY-VALIDATION IS SMZM

ELEMENT DESIGNATOR IS KEY

ELEMENT DESIGNATOR IS VARIABLE

PRESENCE IS PRESENCE-REQUIRED

ELEMENT DEFINITION IS

'THIS FIELD CONTAINS THE BILLET NUMBER WHICH IS ASSIGNED

'BY THE NAVAL INTELLIGENCE COMMAND'

SOURCE-DOC IS NONE

DATA-SUBJECT IS 'SECURITY, GENERAL'

USER IS 'NIC-44'

RESPONSIBLE FOR DEFINITION

COMMENTS

'THE TAPE-FILE SYSTEM USED 3 DIGIT BILLET NUMBERS,

'BUT EXPANSION TO 4 DIGITS IS EXPECTED..'

- ADD ELEMENT NAME IS BILLET-SEC-CCCE

PREPARED BY 'SDC-ISI'

ELEMENT DESCRIPTION IS

'MANAGEMENT NUMBER ASSIGNED A BILLET'

ENTRY-SECURITY IS ENTRY-UNCLASSIFIED

DATA-SECURITY IS DATA-UNCLASSIFIED

APPLICATION-SYSTEM IS SECSS

SERVICE-SUPPORTED IS S8152CC6

PICTURE IS 9(5)

USAGE IS DISPLAY

VALUE IS ZEROS

STORE-VALIDATION IS SSZM

MODIFY-VALIDATION IS SMZM

ELEMENT DESIGNATOR IS VARIABLE

PRESENCE IS PRESENCE-OPTIONAL

ELEMENT DEFINITION IS

'THIS FIELD CONTAINS A 5 DIGIT CODE FOR MANAGEMENT

'PURPOSES FOR OFFICER/ENLISTED BILLETING'

SOURCE-DOC IS NONE

DATA-SUBJECT IS 'SECURITY, GENERAL'

USER IS 'NIC-44'

RESPONSIBLE FOR DEFINITION.

- ADD ELEMENT NAME IS BRANCH-SERVICE

PREPARED BY 'SDC-ISI'

ELEMENT DESCRIPTION IS

'BRANCH OF SERVICE'

ENTRY-SECURITY IS ENTRY-UNCLASSIFIED

DATA-SECURITY IS DATA-UNCLASSIFIED

APPLICATION-SYSTEM IS SECSS

SERVICE-SUPPORTED IS S8152CC4

PICTURE IS X



MEMBER NAME DIESMS  
 USAGE IS DISPLAY  
 VALUE IS SPACES  
 STORE-VALIDATION IS SSZJ  
 MODIFY-VALIDATION IS SMZH  
 ELEMENT DESIGNATOR IS FIXED  
 PRESENCE IS PRESENCE-OPTIONAL  
 ELEMENT DEFINITION IS  
 \*THIS FIELD CONTAINS THE INCUMBENT'S BRANCH OF SERVICE.\*  
 USER IS 'NIC-44'  
 RESPONSIBLE FOR DEFINITION

RANGE IS 'A'  
 RANGE IS 'C'  
 RANGE IS 'F'  
 RANGE IS 'G'  
 RANGE IS 'N'  
 RANGE IS 'M'  
 RANGE IS 'N'  
 RANGE IS 'R'  
 RANGE IS 'R'  
 RANGE IS 'A'  
 RANGE IS 'V'

COMMENT IS  
 \*THIS FIELD IS CODED TO ONE OF THE FOLLOWING VALUES

- : A ARMY  
 - : C CIVIL SERVICE  
 - : F AIR FORCE  
 - : G COAST GUARD  
 - : N CIVILIAN CONTRACTOR  
 - : M MARINE CORPS  
 - : V NAVY  
 - : R CIVILIAN CONSULTANT  
 - : R NAVAL RESERVIST  
 - : X RECORD ON OLD MICFC-FILM ONLY  
 - : V VISITOR

ADD ELEMENT NAME IS COMMAND-CODE

PREPARED BY 'SDC-ISI'  
 ELEMENT DESCRIPTION IS  
 'COMMAND CODE'

ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-UNCLASSIFIED  
 APPLICATION-SYSTEM IS SECSS  
 SERVICE-SUPPORTED IS SB152C14  
 PICTURE IS 916)

USAGE IS DISPLAY  
 VALUE IS ZEROS  
 STORE-VALIDATION IS SSZF  
 MODIFY-VALIDATION IS SMZA  
 ELEMENT DESIGNATOR IS INDEX  
 PRESENCE IS PRESENCE-REQUIRED  
 ELEMENT DEFINITION IS

\*THIS FIELD CONTAINS THE COMMAND CODE.\*

SOURCE-DDC IS NONE  
 DATA-SUBJECT IS 'SECURITY, GENERAL'  
 USER IS 'NIC-44'

RESPONSIBLE FOR DEFINITION

RANGE IS '000001' THRU '999999'.  
 ADD ELEMENT NAME IS DY-ACC-ESTAE  
 PREPARED BY 'SDC-ISI'

MEMBER NAME DIESMS

SAME AS ELEMENT DAY

ELEMENT DESCRIPTION IS

'ACCESS ESTABLISHED DAY'

APPLICATION-SYSTEM IS SECSS

SERVICE-SUPPORTED IS SB152CC4

PRESENCE IS PRESENCE-REQUIRED

ELEMENT DEFINITION IS

'THIS FIELD CONTAINS THE DAY IN WHICH AN ACCESS

'CODE WAS ESTABLISHED.'

DATA-SUBJECT IS 'SECURITY, GENERAL'

USER IS 'NIC-44'

RESPONSIBLE FOR DEFINITION.

ADD ELEMENT NAME IS DY-ACC-TERM

PREPARED BY 'SOC-ISI'

SAME AS ELEMENT DAY

ELEMENT DESCRIPTION IS

'ACCESS DIS-ESTABLISHED DAY'

APPLICATION-SYSTEM IS SECSS

SERVICE-SUPPORTED IS SB152CC4

PRESENCE IS PRESENCE-REQUIRED

ELEMENT DEFINITION IS

'THIS FIELD CONTAINS THE DAY IN WHICH AN ACCESS

'CODE WAS DIS-ESTABLISHED.'

DATA-SUBJECT IS 'SECURITY, GENERAL'

USER IS 'NIC-44'

RESPONSIBLE FOR DEFINITION.

ADD ELEMENT NAME IS DY-ASSIGN-END

PREPARED BY 'SOC-ISI'

SAME AS ELEMENT DAY

ELEMENT DESCRIPTION IS

'ACCESS TERMINATION DAY'

APPLICATION-SYSTEM IS SECSS

SERVICE-SUPPORTED IS SB152CC4

PRESENCE IS PRESENCE-REQUIRED

ELEMENT DEFINITION IS

'THIS FIELD CONTAINS THE DAY IN WHICH USE OF AN ACCESS

'CODE WAS REVOKED (BY DEBRIEFING OR OTHERWISE).'

DATA-SUBJECT IS 'SECURITY, GENERAL'

USER IS 'NIC-44'

RESPONSIBLE FOR DEFINITION.

ADD ELEMENT NAME IS DY-ASSIGN-ESTAB

PREPARED BY 'SOC-ISI'

SAME AS ELEMENT DAY

ELEMENT DESCRIPTION IS

'PERSON ASSIGNED ACCESS DAY'

APPLICATION-SYSTEM IS SECSS

SERVICE-SUPPORTED IS SB152CC4

PRESENCE IS PRESENCE-REQUIRED

ELEMENT DEFINITION IS

'THIS FIELD CONTAINS THE DAY IN WHICH AN ACCESS

'CODE WAS ASSIGNED TO A PERSON.'

DATA-SUBJECT IS 'SECURITY, GENERAL'

USER IS 'NIC-44'

RESPONSIBLE FOR DEFINITION.

ADD ELEMENT NAME IS DY-3ADG-DISE

PREPARED BY 'SOC-ISI'

SAME AS ELEMENT DAY

ELEMENT DESCRIPTION IS

MEMBER NAME DIESMS

\*BADGE DISPOSITION DAY.\*

APPLICATION-SYSTEM IS SECSS

SERVICE-SUPPORTED IS S8152CC4

PRESENCE IS PRESENCE-RECLIFED

ELEMENT DEFINITION IS

\*THIS FIELD INDICATES THE DAY OF THE DISPOSITION (STILL IN

- \*USE, LOST OR DESTROYED) OF A BADGE.\*

DATA-SUBJECT IS \*SECURITY, GENERAL\*

USER IS \*NIC-44\*

RESPONSIBLE FOR DEFINITION.

ADD ELEMENT NAME IS DY-BADGE-ISSUE

PREPARED BY \*SDC-ISI\*

SAME AS ELEMENT DAY

ELEMENT DESCRIPTION IS

\*BADGE ISSUE DAY\*

APPLICATION-SYSTEM IS SECSS

SERVICE-SUPPORTED IS S8152CC1

PRESENCE IS PRESENCE-RECLIFED

ELEMENT DEFINITION IS

\*THIS FIELD CONTAINS THE DAY THAT THE BADGE WAS ISSUED.\*

DATA-SUBJECT IS \*SECURITY, GENERAL\*

USER IS \*NIC-44\*

RESPONSIBLE FOR DEFINITION.

ADD ELEMENT NAME IS DY-BI-COMP

PREPARED BY \*SDC-ISI\*

SAME AS ELEMENT DAY

ELEMENT DESCRIPTION IS

\*BACKGROUND INVESTIGATION COMPLETION DAY\*

APPLICATION-SYSTEM IS SECSS

SERVICE-SUPPORTED IS S8152CC4

PRESENCE IS PRESENCE-RECLIFED

ELEMENT DEFINITION IS

- \*THIS FIELD CONTAINS THE DAY IN WHICH A BACKGROUND

\*INVESTIGATION IS COMPLETED.\*

DATA-SUBJECT IS \*SECURITY, GENERAL\*

USER IS \*NIC-44\*

RESPONSIBLE FOR DEFINITION.

ADD ELEMENT NAME IS DY-BI-RECSTD

PREPARED BY \*SDC-ISI\*

SAME AS ELEMENT DAY

ELEMENT DESCRIPTION IS

\*BACKGROUND INVESTIGATION REQUEST DAY\*

APPLICATION-SYSTEM IS SECSS

SERVICE-SUPPORTED IS S8152CC4

PRESENCE IS PRESENCE-RECLIFED

ELEMENT DEFINITION IS

- \*THIS FIELD CONTAINS THE DAY IN WHICH A BACKGROUND

\*INVESTIGATION IS REQUESTED.\*

DATA-SUBJECT IS \*SECURITY, GENERAL\*

USER IS \*NIC-44\*

RESPONSIBLE FOR DEFINITION.

ADD ELEMENT NAME IS DY-BI-STARTED

PREPARED BY \*SDC-ISI\*

SAME AS ELEMENT DAY

ELEMENT DESCRIPTION IS

\*BACKGROUND INVESTIGATION START DAY\*

APPLICATION-SYSTEM IS SECSS

SERVICE-SUPPORTED IS S8152CC4

MEMBER NAME DIESMS  
 ADD ELEMENT NAME IS C-SEC-NLM  
 PREPARED BY SDC-ISI  
 SAME AS ELEMENT SOCIAL-SECURITY  
 ELEMENT DESCRIPTION IS  
 SOCIAL SECURITY IDENTIFICATION NUMBER  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-UNCLASSIFIED  
 APPLICATION-SYSTEM IS SECSS  
 SERVICE-SUPPORTED IS S8152CG1  
 ELEMENT DESIGNATOR IS KEY  
 ELEMENT DESIGNATOR IS NC-VERIFY  
 PRESENCE IS PRESENCE-REQUIRED  
 ELEMENT DEFINITION IS  
 THIS FIELD IS SOCIAL SECURITY NUMBER ASSIGNED  
 BY THE SOCIAL SECURITY ADMINISTRATION.  
 STANDARD IS DIAADRF  
 SOURCE-DOC IS NONE  
 DATA-SUBJECT IS SECURITY, GENERAL  
 USER IS NIC-44  
 RESPONSIBLE FOR DEFINITION  
 COMMENTS  
 THE SOCIAL SECURITY NUMBER IS USED AS A KEY..  
 ADD ELEMENT NAME IS LOCATION-STATE  
 PREPARED BY SDC-ISI  
 SAME AS ELEMENT STATE-CODE  
 ELEMENT DESCRIPTION IS  
 LOCATION-STATE  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-UNCLASSIFIED  
 APPLICATION-SYSTEM IS SECSS  
 SERVICE-SUPPORTED IS S8152C14  
 ELEMENT DESIGNATOR IS FIXED  
 PRESENCE IS PRESENCE-REQUIRED  
 ELEMENT DEFINITION IS  
 THIS FIELD CONTAINS A TWO CHARACTER CODE TO INDICATE  
 THE STATE IN WHICH AN OBJECT OF ENTITY IS LOCATED.  
 SOURCE-DOC IS NONE  
 STANDARD IS FIPS  
 DATA-SUBJECT IS SECURITY, GENERAL  
 USER IS NIC-44  
 RESPONSIBLE FOR DEFINITION.  
 ADD ELEMENT NAME IS STORAGE-TYPE  
 PREPARED BY SDC-ISI  
 ELEMENT DESCRIPTION IS  
 TYPE OF STORAGE  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-UNCLASSIFIED  
 APPLICATION-SYSTEM IS SECSS  
 SERVICE-SUPPORTED IS S8152C14  
 PICTURE IS X  
 USAGE IS DISPLAY  
 VALUE IS SPACES  
 STORE-VALIDATION IS SSZI  
 MODIFY-VALIDATION IS SMZI  
 ELEMENT DESIGNATOR IS FIXED  
 PRESENCE IS PRESENCE-REQUIRED  
 ELEMENT DEFINITION IS  
 THIS FIELD INDICATES WHETHER THE SPACE HAS OPEN



MEMBER NAME DIISMS

- 'STORAGE WITH NO SECURITY (OPEN SHELVES), CLOSED

- 'STORAGE (LOCKING SECURE FILE CABINET) OR NO STORAGE

- '(WHICH COULD MEAN AN EMPTY ROOM).'

SOURCE-DOC IS NONE

DATA-SUBJECT IS 'SECURITY, GENERAL'

USER IS 'NIC-44'

RESPONSIBLE FOR DEFINITION

RANGE IS 'N'

RANGE IS 'C'

RANGE IS 'O'

COMMENTS

- 'THIS FIELD MUST HAVE ONE OF THE FOLLOWING VALUES

- ' N NONE

- ' C CLOSED

- ' O OPEN

ADD ELEMENT NAME IS TIME-LOG-IN

PREPARED BY 'SDC-ISI'

ELEMENT DESCRIPTION IS

'TIME LOGGED IN'

ENTRY-SECURITY IS ENTRY-UNCLASSIFIED

DATA-SECURITY IS DATA-UNCLASSIFIED

APPLICATION-SYSTEM IS SECSS

SERVICE-SUPPORTED IS S8152C13

PICTURE IS 9(4)

USAGE IS DISPLAY

VALUE IS ZEROES

STORE-VALIDATION IS SSZP

MODIFY-VALIDATION IS SMZA

ELEMENT DESIGNATOR IS FIXED

PRESENCE IS PRESENCE-REQUIRED

ELEMENT DEFINITION IS

- 'THIS FIELD IDENTIFIES THE TIME A VISITOR ENTERED

- 'A BUILDING.'

SOURCE-DOC IS NONE

DATA-SUBJECT IS 'SECURITY, GENERAL'

USER IS 'NIC-44'

RESPONSIBLE FOR DEFINITION

RANGE IS '0000' THRU '2255'

COMMENTS

- 'THIS FIELD CONTAINS THE TIME IN THE FCKM HHMM, WHERE HHMM

- 'IS A 24 HOUR CLOCK FORMAT

- 'I.E. 2200 IS 10 00 P.M.'

ADD ELEMENT NAME IS TIME-LOG-OUT

PREPARED BY 'SDC-ISI'

ELEMENT DESCRIPTION IS

'TIME LOGGED OUT'

ENTRY-SECURITY IS ENTRY-UNCLASSIFIED

DATA-SECURITY IS DATA-UNCLASSIFIED

APPLICATION-SYSTEM IS SECSS

SERVICE-SUPPORTED IS S8152C13

PICTURE IS 9(4)

USAGE IS DISPLAY

VALUE IS ZEROES

STORE-VALIDATION IS SSZP

MODIFY-VALIDATION IS SMZA

ELEMENT DESIGNATOR IS FIXED

PRESENCE IS PRESENCE-REQUIRED

ELEMENT DEFINITION IS



MEMBER NAME DIESMS  
 RANGE IS 'N'.  
 RANGE IS 'S'.  
 ADD ELEMENT NAME IS LATITUDE  
 PREPARED BY 'SDC-ISI'  
 ELEMENT DESCRIPTION IS  
 'LATITUDE'  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-UNCLASSIFIED  
 ELEMENT DESIGNATOR IS DES-VARIABLE  
 PRESENCE IS PRESENCE-OPTIONAL  
 ELEMENT DEFINITION IS  
 'THE LATITUDE IS THE ANGULAR DISTANCE FROM THE EQUATOR'  
 SUBORDINATE ELEMENTS ARE  
 LATITUDE-DEGREES  
 LATITUDE-MINUTES  
 LATITUDE-SECONDS  
 LATITUDE-HEMISPHERE.  
 ADD ELEMENT NAME IS LONGITUDE-DEGREES  
 PREPARED BY LET  
 ELEMENT DESCRIPTION IS  
 'DEGREES OF LONGITUDE'  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-UNCLASSIFIED  
 APPLICATION-SYSTEM IS UCB  
 PICTURE IS 999  
 USAGE IS DISPLAY  
 VALUE IS ZEROS  
 STORE-VALIDATION IS SSZM  
 MODIFY-VALIDATION IS SMZC  
 ELEMENT DESIGNATOR IS DES-VARIABLE  
 PRESENCE IS PRESENCE-OPTIONAL  
 ELEMENT DEFINITION IS  
 'THE ANGULAR DISTANCE EAST OR WEST FROM THE PRIME OR  
 - 'ZERO MERIDIAN TO A POINT ON THE EARTH'S SURFACE,  
 - 'MEASURED IN DEGREES, FROM 00 DEGREES AT THE PRIME OR  
 - 'ZERO MERIDIAN UP TO, BUT NOT EXCEEDING THE 180 DEGREE  
 - 'ANGLES EAST AND WEST BETWEEN THE PRIME OR ZERO MERIDIAN  
 - 'AND THE 180 DEGREE MERIDIAN'  
 STANDARD IS DIADREF  
 DATA-SUBJECT IS COMMON-ELEMENT  
 USER IS 'NIPSSAOSON' 'ICWNER'  
 RESPONSIBLE FOR DEFINITION  
 RANGE IS '000' THRU '180'.  
 ADD ELEMENT NAME IS LONGITUDE-MINUTES  
 PREPARED BY LET  
 ELEMENT DESCRIPTION IS  
 'MINUTES OF LONGITUDE'  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-UNCLASSIFIED  
 APPLICATION-SYSTEM IS UCB  
 PICTURE IS 99  
 VALUE IS ZEROS  
 STORE-VALIDATION IS SSZM  
 MODIFY-VALIDATION IS SMZC  
 ELEMENT DESIGNATOR IS DES-FIXED  
 PRESENCE IS PRESENCE-OPTIONAL  
 ELEMENT DEFINITION IS  
 'ONE OF 60 EQUAL PARTS OF 1 DEGREE OF ANGLE, THE TOTAL

MEMBER NAME DIESMS  
 - 'OF 60 SUCH PARTS EXPRESSED AS 1 DEGREE OF ANGLE'  
 STANDARD IS DIAADERF  
 USER IS 'NIPSSAO3DN' TOWNER'  
 RESPONSIBLE FOR DEFINITION  
 RANGE IS '00' THRU '99'  
 ADD ELEMENT NAME IS LONGITUDE-SECONDS  
 PREPARED BY LET  
 ELEMENT DESCRIPTION IS  
 'SECONDS OF LONGITUDE'  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-UNCLASSIFIED  
 APPLICATION-SYSTEM IS UCB  
 PICTURE IS 99  
 USAGE IS DISPLAY  
 VALUE IS ZEROS  
 STORE-VALIDATION IS SSZM  
 MODIFY-VALIDATION IS SMZC  
 ELEMENT DESIGNATOR IS DES-FIXED  
 PRESENCE IS PRESENCE-OPTICAL  
 ELEMENT DEFINITION IS  
 'ONE OF 60 EQUAL PARTS OF A MINUTE OF ANGLE, THE TOTAL  
 - 'OF 60 SUCH PARTS EXPRESSED AS 1 MINUTE OF ANGLE'  
 STANDARD IS DIAADERF  
 DATA-SUBJECT IS COMMON-ELEMENT  
 USER IS 'NIPSSAO3DN' TOWNER'  
 RESPONSIBLE FOR DEFINITION  
 RANGE IS '00' THRU '99'  
 ADD ELEMENT NAME IS LONGITUDE-HEMISPHERE  
 PREPARED BY LET  
 ELEMENT DESCRIPTION IS  
 'HEMISPHERE OF LONGITUDE'  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-UNCLASSIFIED  
 APPLICATION-SYSTEM IS UCB  
 PICTURE IS X  
 USAGE IS DISPLAY  
 VALUE IS SPACES  
 STORE-VALIDATION IS SSZG  
 MODIFY-VALIDATION IS SMZA  
 ELEMENT DESIGNATOR IS DES-CCDE  
 PRESENCE IS PRESENCE-OPTICAL  
 ELEMENT DEFINITION IS  
 'EAST OR WEST'  
 DATA-SUBJECT IS COMMON-ELEMENT  
 USER IS 'NIPSSAO3DN' TOWNER'  
 RESPONSIBLE FOR DEFINITION  
 RANGE IS 'E'  
 RANGE IS 'W'  
 ADD ELEMENT NAME IS LONGITUDE  
 PREPARED BY 'SDC-ISI'  
 ELEMENT DESCRIPTION IS  
 'LONGITUDE'  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-UNCLASSIFIED  
 ELEMENT DESIGNATOR IS DES-VARIABLE  
 PRESENCE IS PRESENCE-OPTICAL  
 ELEMENT DEFINITION IS  
 'THE LONGITUDE IS THE ANGULAR DISTANCE FROM PRIME MERIDIAN'

MEMBER NAME DIESMS  
SUBORDINATE ELEMENTS ARE  
LONGITUDE-DEGREES  
LONGITUDE-MINUTES  
LONGITUDE-SECONDS  
LONGITUDE-HEMISPHERE.  
ADD ELEMENT NAME IS ALTITUDE  
PREPARED BY 'SDC-ISI'  
ELEMENT DESCRIPTION IS  
'ALTITUDE'  
ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
DATA-SECURITY IS DATA-UNCLASSIFIED  
APPLICATION-SYSTEM IS 'SECSS'  
SERVICE-SUPPORTED IS S8152CC6  
PICTURE IS S9(5)  
USAGE IS DISPLAY  
VALUE IS ZEROS  
STORE-VALIDATION IS SSZG  
MODIFY-VALIDATION IS SMZA  
ELEMENT DESIGNATOR IS DES-VARIABLE  
PRESENCE IS PRESENCE-OPTIONAL  
ELEMENT DEFINITION IS  
'THE ALTITUDE IS THE VERTICAL ELEVATION ABOVE SEA LEVEL'  
DATA-SUBJECT IS 'SECURITY, GENERAL'  
USER IS 'NIC-44'  
RESPONSIBLE FOR DEFINITION.  
ADD ELEMENT NAME IS LOCATION-CNTRY  
PREPARED BY 'SDC-ISI'  
SAME AS ELEMENT COUNTRY-CODE  
ELEMENT DESCRIPTION IS  
'COUNTRY CODE'  
APPLICATION-SYSTEM IS 'SECSS'  
SERVICE-SUPPORTED IS S8152CC6  
USAGE IS DISPLAY  
VALUE IS SPACES  
STORE-VALIDATION IS SSZG  
MODIFY-VALIDATION IS SMZA  
ELEMENT DESIGNATOR IS DES-VARIABLE  
PRESENCE IS PRESENCE-OPTIONAL  
ELEMENT DEFINITION IS  
'THIS FIELD IS A TWO CHARACTER ALPHANUMERIC CODE TO  
- INDICATE THE COUNTRY IN WHICH AN OBJECT OR ENTITY  
- IS LOCATED.'  
STANDARD IS DIAADREF  
USER IS 'NIC-44'  
RESPONSIBLE FOR DEFINITION  
COMMENTS  
'SECONDARY INDEX'  
ADD ELEMENT NAME IS LOC-ZIP-CODE  
PREPARED BY 'SDC-ISI'  
ELEMENT DESCRIPTION IS  
'ZIP CODE'  
ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
DATA-SECURITY IS DATA-UNCLASSIFIED  
APPLICATION-SYSTEM IS 'SECSS'  
SERVICE-SUPPORTED IS S8152CC6  
PICTURE IS X(9)  
USAGE IS DISPLAY  
VALUE IS SPACES

MEMBER NAME DIES45  
 STORE-VALIDATION IS SSZG  
 MODIFY-VALIDATION IS SMZA  
 ELEMENT DESIGNATOR IS DES-VARIABLE  
 PRESENCE IS PRESENCE-OPTIONAL  
 ELEMENT DEFINITION IS  
 'ZIP CODE IS A CODE ASSIGNED TO POST OFFICE SERVICE AREAS'  
 DATA-SUBJECT IS 'SECURITY, GENERAL'  
 USER IS 'NIC-44'  
 RESPONSIBLE FOR DEFINITION.  
 ADD ELEMENT NAME IS LOC-NAME-TYPE  
 PREPARED BY 'SDC-ISI'  
 ELEMENT DESCRIPTION IS  
 'LOCATION NAME/TYPE'  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-UNCLASSIFIED  
 ELEMENT DESIGNATOR IS DES-VARIABLE  
 ELEMENT DEFINITION IS  
 'THIS FIELD CONTAINS THE NAME AND TYPE OF THE OBJECT/ENTITY'  
 SUBORDINATE ELEMENTS ARE  
 LOCATION-NAME  
 LOCATION-TYPE.  
 ADD ELEMENT NAME IS STATE-ZIP-CNTRY  
 PREPARED BY 'SDC-ISI'  
 ELEMENT DESCRIPTION IS  
 'STATE, ZIP, AND COUNTRY CODES'  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-UNCLASSIFIED  
 ELEMENT DESIGNATOR IS DES-VARIABLE  
 ELEMENT DEFINITION IS  
 'THIS FIELD CONTAINS THE STATE CODE, ZIP CODE, AND THE  
 - 'COUNTRY CODE OF AN ENTITY'  
 SUBORDINATE ELEMENTS ARE  
 LOCATION-STATE  
 LOCATION-CNTRY  
 LOC-ZIP-CODE.  
 ADD ELEMENT NAME IS LAT-LONG-ALT  
 PREPARED BY 'SDC-ISI'  
 ELEMENT DESCRIPTION IS  
 'LATITUDE/LONGITUDE/ALTITUDE'  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-UNCLASSIFIED  
 ELEMENT DESIGNATOR IS DES-VARIABLE  
 ELEMENT DEFINITION IS  
 'THIS FIELD CONTAINS THE LOCATION OF AN OBJECT OR ENTITY  
 - 'IN TERMS OF LATITUDE, LONGITUDE, AND ALTITUDE'  
 SUBORDINATE ELEMENTS ARE  
 LATITUDE  
 LONGITUDE  
 ALTITUDE.  
 ADD ELEMENT NAME IS ACCESS-FLCAT  
 PREPARED BY 'SDC-ISI'  
 ELEMENT DESCRIPTION IS  
 'FLCATING ACCESS INFORMATION'  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-UNCLASSIFIED  
 ELEMENT DESIGNATOR IS FIXED  
 ELEMENT DEFINITION IS  
 'THIS FIELD CONTAINS THE TOTALS THAT CONTROL ACCESS'



```

MEMBER NAME DIESMS
- 'CODES FOR FLOATING ALLOCATION.'
  SUBORDINATE ELEMENTS ARE
  TOT-ACCESS-AUTH
  TOT-ACCESS-FILL
  TOT-ACCESS-FLOAT
  TOT-AUTHZD-FLOAT
  TOT-BILT-READOUT.
ADD ELEMENT NAME IS BADGE-DISP-INFO
  PREPARED BY 'SDC-ISI'
  ELEMENT DESCRIPTION IS
    'BADGE DISPOSITION INFORMATION'
  ENTRY-SECURITY IS ENTRY-UNCLASSIFIED
  DATA-SECURITY IS DATA-UNCLASSIFIED
  ELEMENT DESIGNATOR IS FIXED
  ELEMENT DEFINITION IS
    'THIS FIELD CONTAINS THE BADGE DISPOSITION
    - INFORMATION.'
  SUBORDINATE ELEMENTS ARE
  BADG-DISPOSITION
  DATE-BADG-DISP.
ADD ELEMENT NAME IS BADGE-TYPE-CODE
  PREPARED BY 'SDC-ISI'
  ELEMENT DESCRIPTION IS
    'BADGE TYPE INFORMATION'
  ENTRY-SECURITY IS ENTRY-UNCLASSIFIED
  DATA-SECURITY IS DATA-UNCLASSIFIED
  ELEMENT DESIGNATOR IS FIXED
  ELEMENT DEFINITION IS
    'THIS FIELD CONTAINS THE FLAGS AND INDICATORS THAT
    - DESCRIBE A BADGE.'
  SUBORDINATE ELEMENTS ARE
  BADGE-TYPE
  SECURE-IC-CODE.
ADD ELEMENT NAME IS BILLET-FLAGS
  PREPARED BY 'SDC-ISI'
  ELEMENT DESCRIPTION IS
    'BILLET FLAG INFORMATION'
  ENTRY-SECURITY IS ENTRY-UNCLASSIFIED
  DATA-SECURITY IS DATA-UNCLASSIFIED
  ELEMENT DESIGNATOR IS FIXED
  ELEMENT DEFINITION IS
    'THIS FIELD CONTAINS THE FLAGS THAT CONTROL AND INDICATE
    - BILLET USAGE.'
  SUBORDINATE ELEMENTS ARE
  FLG-BILT-TYPE
  FLG-BILT-STATUS
  FLG-BE-READ-OUT
  FLG-USER-CODE
  FLG-SAO-CODE.
ADD ELEMENT NAME IS DATE-ACC-ESTAE
  PREPARED BY 'SDC-ISI'
  SAME AS ELEMENT DATE-COMM
  ELEMENT DESCRIPTION IS
    'ACCESS ESTABLISHED DATE'
  ENTRY-SECURITY IS ENTRY-UNCLASSIFIED
  DATA-SECURITY IS DATA-UNCLASSIFIED
  ELEMENT DESIGNATOR IS FIXED
  ELEMENT DEFINITION IS

```



MEMBER NAME DIESMS

- 'THIS FIELD CONTAINS THE DATE IN WHICH AN ACCESS  
'CODE WAS ESTABLISHED.'  
SUBORDINATE ELEMENTS ARE  
YR-ACC-ESTAB  
MO-ACC-ESTAB  
DY-ACC-ESTAB.

ADD ELEMENT NAME IS DATE-ACC-TERM  
PREPARED BY 'SDC-ISI'  
SAME AS ELEMENT DATE-COMMCA  
ELEMENT DESCRIPTION IS  
'ACCESS TERMINATION DATE'  
ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
DATA-SECURITY IS DATA-UNCLASSIFIED  
ELEMENT DESIGNATOR IS FIXED  
ELEMENT DEFINITION IS

- 'THIS FIELD CONTAINS THE DATE IN WHICH AN ACCESS  
'CODE WAS TERMINATED.'  
SUBORDINATE ELEMENTS ARE  
YR-ACC-TERM  
MO-ACC-TERM  
DY-ACC-TERM.

ADD ELEMENT NAME IS DATE-ASSIGN-END  
PREPARED BY 'SDC-ISI'  
SAME AS ELEMENT DATE-COMMCA  
ELEMENT DESCRIPTION IS  
'ACCESS TERMINATION DATE'  
ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
DATA-SECURITY IS DATA-UNCLASSIFIED  
ELEMENT DESIGNATOR IS FIXED  
ELEMENT DEFINITION IS

- 'THIS FIELD CONTAINS THE DATE IN WHICH USE OF AN ACCESS  
'CODE WAS REVOKED (BY DEBRIEFING OR OTHERWISE).'  
SUBORDINATE ELEMENTS ARE  
YR-ASSIGN-END  
MO-ASSIGN-END  
DY-ASSIGN-END.

ADD ELEMENT NAME IS DATE-ASSIGN-ESTAB  
PREPARED BY 'SDC-ISI'  
SAME AS ELEMENT DATE-COMMCA  
ELEMENT DESCRIPTION IS  
'PERSON ASSIGNED ACCESS DATE'  
ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
DATA-SECURITY IS DATA-UNCLASSIFIED  
ELEMENT DESIGNATOR IS FIXED  
ELEMENT DEFINITION IS

- 'THIS FIELD CONTAINS THE DATE IN WHICH AN ACCESS  
'CODE WAS ASSIGNED TO A PERSON.'  
SUBORDINATE ELEMENTS ARE  
YR-ASSIGN-ESTAB  
MO-ASSIGN-ESTAB  
DY-ASSIGN-ESTAB.

ADD ELEMENT NAME IS DATE-BADG-DISF  
PREPARED BY 'SDC-ISI'  
SAME AS ELEMENT DATE-COMMCA  
ELEMENT DESCRIPTION IS  
'DISPOSITION DATE'  
ELEMENT DEFINITION IS  
'THIS FIELD INDICATES THE DATE OF THE DISPOSITION (STILL IN

MEMBER NAME DIFSMS  
 - 'USE, LOST OR DESTROYED) OF A BADGE.'  
 SUBORDINATE ELEMENTS ARE  
 YR-BADG-DISP  
 MO-BADG-DISP  
 DY-BADG-DISP.  
 ADD ELEMENT NAME IS DATE-BADGE-ISSUE  
 PREPARED BY 'SDC-ISI'  
 SAME AS ELEMENT DATE-COMMCN  
 ELEMENT DESCRIPTION IS  
 'BADGE ISSUE DATE'  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-UNCLASSIFIED  
 ELEMENT DESIGNATOR IS FIXED  
 ELEMENT DEFINITION IS  
 'THIS FIELD CONTAINS THE DATE THAT THE BADGE WAS ISSUED.'  
 SUBORDINATE ELEMENTS ARE  
 YR-BADGE-ISSUE  
 MO-BADGE-ISSUE  
 DY-BADGE-ISSUE.  
 ADD ELEMENT NAME IS DATE-BI-CCMP  
 PREPARED BY 'SDC-ISI'  
 SAME AS ELEMENT DATE-COMMCN  
 ELEMENT DESCRIPTION IS  
 'BACKGROUND INVESTIGATION COMPLETION DATE'  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-UNCLASSIFIED  
 ELEMENT DESIGNATOR IS FIXED  
 ELEMENT DEFINITION IS  
 'THIS FIELD CONTAINS THE DATE IN WHICH A BACKGROUND  
 - INVESTIGATION IS COMPLETED FOR A PERSON.'  
 SUBORDINATE ELEMENTS ARE  
 YR-BI-CCMP  
 MO-BI-CCMP  
 DY-BI-CCMP.  
 ADD ELEMENT NAME IS DATE-BI-REQSTD  
 PREPARED BY 'SDC-ISI'  
 SAME AS ELEMENT DATE-COMMCN  
 ELEMENT DESCRIPTION IS  
 'BACKGROUND INVESTIGATION REQUEST DATE'  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-UNCLASSIFIED  
 ELEMENT DESIGNATOR IS FIXED  
 ELEMENT DEFINITION IS  
 'THIS FIELD CONTAINS THE DATE ON WHICH A BACKGROUND  
 - INVESTIGATION IS REQUESTED FOR A PERSON.'  
 SUBORDINATE ELEMENTS ARE  
 YR-BI-REQSTD  
 MO-BI-REQSTD  
 DY-BI-REQSTD.  
 ADD ELEMENT NAME IS DATE-BI-STARTED  
 PREPARED BY 'SDC-ISI'  
 SAME AS ELEMENT DATE-COMMCN  
 ELEMENT DESCRIPTION IS  
 'BACKGROUND INVESTIGATION START DATE'  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-UNCLASSIFIED  
 ELEMENT DESIGNATOR IS FIXED  
 ELEMENT DEFINITION IS

MEMBER NAME DIESMS

- 'THIS FIELD CONTAINS THE DATE ON WHICH A BACKGROUND  
INVESTIGATION IS STARTED FOR A PERSON.'

SUBORDINATE ELEMENTS ARE

YR-BI-STARTED

MO-BI-STARTED

DY-BI-STARTED.

ADD ELEMENT NAME IS DATE-BILT-CREATE

PREPARED BY 'SOC-ISI'

SAME AS ELEMENT DATE-COMMON

ELEMENT DESCRIPTION IS

'DATE A BILLET WAS CREATED'

ENTRY-SECURITY IS ENTRY-UNCLASSIFIED

DATA-SECURITY IS DATA-UNCLASSIFIED

ELEMENT DESIGNATOR IS FIXED

ELEMENT DEFINITION IS

- 'THIS FIELD CONTAINS THE DATE THAT A BILLET WAS  
CREATED.'

SUBORDINATE ELEMENTS ARE

YR-BILT-CREATD

MO-BILT-CREATD

DY-BILT-CREATD.

ADD ELEMENT NAME IS DATE-CERT-EXPIRE

PREPARED BY 'SOC-ISI'

SAME AS ELEMENT DATE-COMMON

ELEMENT DESCRIPTION IS

'CERTIFICATION EXPIRE DATE'

ENTRY-SECURITY IS ENTRY-UNCLASSIFIED

DATA-SECURITY IS DATA-UNCLASSIFIED

ELEMENT DESIGNATOR IS FIXED

ELEMENT DEFINITION IS

- 'THIS FIELD CONTAINS THE EXPIRATION DATE FOR  
PERSONNEL CERTIFIED TO WORK IN A BILLET.'

SUBORDINATE ELEMENTS ARE

YR-CERT-EXPIRE

MO-CERT-EXPIRE

DY-CERT-EXPIRE.

ADD ELEMENT NAME IS DATE-FINAL-ACCRED

PREPARED BY 'SOC-ISI'

SAME AS ELEMENT DATE-COMMON

ELEMENT DESCRIPTION IS

'FINAL ACCREDITATION DATE'

ENTRY-SECURITY IS ENTRY-UNCLASSIFIED

DATA-SECURITY IS DATA-UNCLASSIFIED

ELEMENT DESIGNATOR IS FIXED

ELEMENT DEFINITION IS

- 'THIS FIELD CONTAINS THE DATE WHEN NIC-44  
ACCREDITED THE SPACE FOR THE SPECIFIED LEVELS.'

SUBORDINATE ELEMENTS ARE

YR-FINAL-ACCRED

MO-FINAL-ACCRED

DY-FINAL-ACCRED.

ADD ELEMENT NAME IS DATE-INSPECTED

PREPARED BY 'SOC-ISI'

SAME AS ELEMENT DATE-COMMON

ELEMENT DESCRIPTION IS

'DATE LAST INSPECTED'

ENTRY-SECURITY IS ENTRY-UNCLASSIFIED

DATA-SECURITY IS DATA-UNCLASSIFIED

MEMBER NAME DITSM5

ELEMENT DESIGNATOR IS FIXED

ELEMENT DEFINITION IS

- 'THIS FIELD CONTAINS THE DATE OF THE LAST

'INSPECTION OF A SPACE.'

SUBORDINATE ELEMENTS ARE

YR-INSPECTED

MO-INSPECTED

ADD DY-INSPECTED.

ELEMENT NAME IS DATE-INTR-ACCRED

PREPARED BY 'SDC-ISI'

SAME AS ELEMENT DATE-COMMON

ELEMENT DESCRIPTION IS

'INTERIM ACCREDITATION DATE'

ENTRY-SECURITY IS ENTRY-UNCLASSIFIED

DATA-SECURITY IS DATA-UNCLASSIFIED

ELEMENT DESIGNATOR IS FIXED

ELEMENT DEFINITION IS

- 'THIS FIELD CONTAINS THE DATE THAT A SPACE WAS GIVEN

- 'AN INTERIM ACCREDITATION. THE SPACE HAS THIS

- 'ACCREDITATION UNTIL A TECHNICAL SURVEILLANCE COUNTER

'MEASURES SURVEY IS COMPLETED.'

SUBORDINATE ELEMENTS ARE

YR-INTER-ACCRED

MO-INTER-ACCRED

ADD DY-INTER-ACCRED.

ELEMENT NAME IS DATE-LAST-ACCRED

PREPARED BY 'SDC-ISI'

SAME AS ELEMENT DATE-COMMON

ELEMENT DESCRIPTION IS

'LAST ACCREDITATION DATE'

ENTRY-SECURITY IS ENTRY-UNCLASSIFIED

DATA-SECURITY IS DATA-UNCLASSIFIED

ELEMENT DESIGNATOR IS FIXED

ELEMENT DEFINITION IS

- 'THIS FIELD CONTAINS THE DATE ON WHICH A SPACE WAS LAST

- 'ACCREDITED.'

SUBORDINATE ELEMENTS ARE

YR-LAST-ACCRED

MO-LAST-ACCRED

ADD DY-LAST-ACCRED.

ELEMENT NAME IS DATE-LOG-IN

PREPARED BY 'SDC-ISI'

SAME AS DATE-COMMON

ELEMENT DESCRIPTION IS

'DATE LOGGED IN'

ENTRY-SECURITY IS ENTRY-UNCLASSIFIED

DATA-SECURITY IS DATA-UNCLASSIFIED

ELEMENT DESIGNATOR IS FIXED

ELEMENT DEFINITION IS

- 'THIS DATE FIELD IDENTIFIES THE DATE A VISITOR ENTERED

- 'A BUILDING.'

SUBORDINATE ELEMENTS ARE

YR-LOG-IN

MO-LOG-IN

ADD DY-LOG-IN.

ELEMENT NAME IS DATE-LOG-CLT

PREPARED BY 'SDC-ISI'

SAME AS ELEMENT DATE-COMMON



MEMBER NAME DIESMS

ELEMENT DESCRIPTION IS  
 'DATE LOGGED OUT'  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-UNCLASSIFIED  
 ELEMENT DESIGNATOR IS FIXED  
 ELEMENT DEFINITION IS  
 'THIS FIELD IDENTIFIES THE DATE A VISITOR LEFT  
 - 'A BUILDING.'  
 SUBORDINATE ELEMENTS ARE  
 YR-LOG-OUT  
 MO-LOG-OUT  
 DY-LOG-OUT

ADD ELEMENT NAME IS DATE-READ-IN  
 PREPARED BY 'SOC-ISI'  
 SAME AS ELEMENT DATE-COMMON  
 ELEMENT DESCRIPTION IS  
 'DATE READ INTO A BILLET'  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-UNCLASSIFIED  
 ELEMENT DESIGNATOR IS FIXED  
 ELEMENT DEFINITION IS  
 'THIS FIELD CONTAINS THE DATE THAT AN INCUMBENT WAS  
 - 'READ INTO A BILLET.'  
 SUBORDINATE ELEMENTS ARE  
 YR-READ-IN  
 MO-READ-IN  
 DY-READ-IN

ADD ELEMENT NAME IS DATE-READ-OUT  
 PREPARED BY 'SOC-ISI'  
 SAME AS ELEMENT DATE-COMMON  
 ELEMENT DESCRIPTION IS  
 'FLAGGED TO BE READ OUT DATE'  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-UNCLASSIFIED  
 ELEMENT DESIGNATOR IS FIXED  
 ELEMENT DEFINITION IS  
 'THIS FIELD CONTAINS THE DATE THAT A BILLET WAS FLAGGED TO  
 - 'HAVE ITS INCUMBENT READ OUT.'  
 SUBORDINATE ELEMENTS ARE  
 YR-READ-OUT  
 MO-READ-OUT  
 DY-READ-OUT

ADD ELEMENT NAME IS DATE-SMS-CMTS  
 PREPARED BY 'SOC-ISI'  
 SAME AS ELEMENT DATE-COMMON  
 ELEMENT DESCRIPTION IS  
 'DATE OF INSPECTION COMMENTS'  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-UNCLASSIFIED  
 ELEMENT DESIGNATOR IS FIXED  
 ELEMENT DEFINITION IS  
 'THIS FIELD CONTAINS THE DATE FOR THE COMMENTS OF THE  
 - 'INSPECTION OF A SPACE. IT IS USED AS ONE OF TWO KEY  
 - 'FIELDS TO ACCESS THE DATA BASE COMMENTS RECORD.'  
 SUBORDINATE ELEMENTS ARE  
 YR-SMS-CMTS  
 MO-SMS-CMTS  
 DY-SMS-CMTS



MEMBER NAME DIESMS  
 ADD ELEMENT NAME IS DATE-TSCM-COMF  
 PREPARED BY 'SDC-ISI'  
 SAME AS ELEMENT DATE-COMF  
 ELEMENT DESCRIPTION IS  
 'TSCM COMPLETION DATE'  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-UNCLASSIFIED  
 ELEMENT DESIGNATOR IS FIXED  
 ELEMENT DEFINITION IS  
 'THIS FIELD CONTAINS THE DATE THAT NAVAL INVESTIGATIVE  
 - SERVICES COMPLETED THE TECHNICAL SURVEILLANCE COUNTER  
 - MEASURES SURVEY.'  
 SUBORDINATE ELEMENTS ARE  
 YR-TSCM-COMP  
 MO-TSCM-COMP  
 DY-TSCM-COMP  
 ADD ELEMENT NAME IS DATE-TSCM-REQ  
 PREPARED BY 'SDC-ISI'  
 SAME AS ELEMENT DATE-COMF  
 ELEMENT DESCRIPTION IS  
 'TSCM REQUEST DATE'  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-UNCLASSIFIED  
 ELEMENT DESIGNATOR IS FIXED  
 ELEMENT DEFINITION IS  
 'THIS FIELD CONTAINS THE DATE A REQUEST WAS MADE TO NAVAL  
 - INVESTIGATIVE SERVICE (NIS) FOR PERFORMING A TECHNICAL  
 - SURVEILLANCE COUNTER MEASURES SURVEY (INSPECTION OF THE  
 - SPACE TO BE SURE MINIMUM REQUIREMENTS FOR THE HIGHEST  
 LEVEL OF CLEARANCE ARE MET).'  
 SUBORDINATE ELEMENTS ARE  
 YR-TSCM-REQ  
 MO-TSCM-REQ  
 DY-TSCM-REQ  
 ADD ELEMENT NAME IS DTG-LOG-IN  
 PREPARED BY 'SDC-ISI'  
 ELEMENT DESCRIPTION IS  
 'DATE AND TIME LOGGED IN'  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-UNCLASSIFIED  
 ELEMENT DESIGNATOR IS FIXED  
 ELEMENT DEFINITION IS  
 'THIS GROUP COMBINES THE DAY AND TIME THAT A VISITOR  
 - ENTERED (LOGGED INTO) A BUILDING.'  
 SUBORDINATE ELEMENTS ARE  
 DATE-LOG-IN  
 TIME-LOG-IN  
 ADD ELEMENT NAME IS DTG-LOG-OUT  
 PREPARED BY 'SDC-ISI'  
 ELEMENT DESCRIPTION IS  
 'DAY AND TIME LOGGED OUT'  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-UNCLASSIFIED  
 ELEMENT DESIGNATOR IS FIXED  
 ELEMENT DEFINITION IS  
 'THIS GROUP COMBINES THE DAY AND TIME THAT A VISITOR  
 - LEFT (LOGGED OUT OF) A BUILDING.'  
 SUBORDINATE ELEMENTS ARE

MEMBER NAME DIESMS  
 BY-LOG-CUT  
 TIME-LOG-OUT.  
 ADD ELEMENT NAME IS RANK-SERVICE  
 PREPARED BY 'SOC-ISI'  
 ELEMENT DESCRIPTION IS  
 'RANK AND SERVICE INFORMATION'  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-UNCLASSIFIED  
 ELEMENT DESIGNATOR IS FIXED  
 ELEMENT DEFINITION IS  
 'THIS GROUP CONTAINS THE FIELDS THAT CONTAIN RANK AND  
 'SERVICE INFORMATION.'  
 SUBORDINATE ELEMENTS ARE  
 RANK-OR-GRADE  
 BRANCH-SERVICE.  
 ADD ELEMENT NAME IS ID-LOCATION  
 PREPARED BY LFT  
 ELEMENT DESCRIPTION IS  
 'LOCATION OF IDENTIFIER'  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-UNCLASSIFIED  
 APPLICATION-SYSTEM IS 'SECSS'  
 SERVICE-SUPPORTED IS 'S8152CC6'  
 PICTURE IS X(30)  
 USAGE IS DISPLAY  
 VALUE IS SPACES  
 STORE-VALIDATION IS SSZA  
 MODIFY-VALIDATION IS SMZG  
 ELEMENT DESIGNATOR IS KEY  
 PRESENCE IS PRESENCE-REQUIRED  
 ELEMENT DEFINITION IS  
 'THE IDENTIFIER OF THE ELEMENT'  
 DATA-SUBJECT IS 'SECURITY, GENERAL'  
 USER IS 'NIC-44'  
 RESPONSIBLE FOR DEFINITION  
 SUBORDINATE ELEMENTS ARE  
 LOCATION-ID  
 COMMENTS  
 RECORD CALC KEY.  
 ADD ELEMENT NAME IS DA-LOCATION  
 PREPARED BY LET  
 ELEMENT DESCRIPTION IS  
 'LOCATION OF IDENTIFIER'  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-UNCLASSIFIED  
 APPLICATION-SYSTEM IS 'SECSS'  
 SERVICE-SUPPORTED IS 'S8152CC6'  
 PICTURE IS X(30)  
 USAGE IS DISPLAY  
 VALUE IS SPACES  
 STORE-VALIDATION IS SSZA  
 MODIFY-VALIDATION IS SMZG  
 ELEMENT DESIGNATOR IS KEY  
 PRESENCE IS PRESENCE-REQUIRED  
 ELEMENT DEFINITION IS  
 'THE IDENTIFIER OF THE ELEMENT'  
 DATA-SUBJECT IS 'SECURITY, GENERAL'  
 USER IS 'NIC-44'

MEMBER NAME DIESMS  
 RESPONSIBLE FOR DEFINITION  
 COMMENTS  
 SUBORDINATE ELEMENTS ARE  
 LOC-NAME-TYPE  
 LOCATION-SIZE  
 UNIT-OF-MEASURE  
 LAT-LONG-ALT  
 STATE-ZIP-ENTRY.

ADD ELEMENT NAME IS DA-ORGAN-BILLET  
 PREPARED BY LET  
 ELEMENT DESCRIPTION IS  
 'ORGANIZATION BILLET DATA'  
 APPLICATION-SYSTEM IS 'SECSS'  
 PRESENCE IS PRESENCE-REQUIRED  
 ELEMENT DEFINITION IS  
 - 'THE FIELD IS THE CODE WHICH DEFINES THE RELATIONSHIP'  
 'BETWEEN ORGANIZATIONS AND BILLETS'  
 DATA-SUBJECT IS 'SECURITY, GENERAL'  
 USER IS 'NIC-44'  
 RESPONSIBLE FOR DEFINITION  
 SUBORDINATE ELEMENTS ARE  
 FLG-SSO-CODE.

ADD ELEMENT NAME IS CMD-CODE  
 PREPARED BY LET  
 SAME AS ELEMENT COMMAND-CODE  
 ELEMENT DESCRIPTION IS  
 'SMS COMMAND IDENT CODE'  
 APPLICATION-SYSTEM IS 'SECSS'  
 PRESENCE IS PRESENCE-OPTIONAL  
 ELEMENT DEFINITION IS  
 - 'THE CODE USED BY THE SECURITY MANAGEMENT'  
 'SUBSYSTEM FOR COMMAND IDENTIFICATION'  
 DATA-SUBJECT IS 'SECURITY, GENERAL'  
 USER IS 'NIC-44'  
 RESPONSIBLE FOR DEFINITION  
 RANGE IS '000000' THRU '999999'  
 COMMENTS  
 'SECONDARY INDEX'.

ADD ELEMENT NAME IS ID-LOCATION-ACCESS  
 PREPARED BY LET  
 ELEMENT DESCRIPTION IS  
 'LOCATION ACCESS IDENTIFIER'  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-UNCLASSIFIED  
 APPLICATION-SYSTEM IS 'SECSS'  
 ELEMENT DESIGNATOR IS DES-KEY  
 PRESENCE IS PRESENCE-REQUIRED  
 ELEMENT DEFINITION IS  
 'LOCATION ACCESS IDENTIFICATION FIELDS'  
 DATA-SUBJECT IS 'SECURITY, GENERAL'  
 USER IS 'NIC-44'  
 RESPONSIBLE FOR DEFINITION  
 SUBORDINATE ELEMENTS ARE  
 DATE-INSPECTED.

ADD ELEMENT NAME IS DA-LOCATION-ACCESS  
 PREPARED BY LET  
 ELEMENT DESCRIPTION IS  
 'LOCATION ACCESS DATA'

MEMBER NAME DIESMS  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-UNCLASSIFIED  
 APPLICATION-SYSTEM IS 'SECSS'  
 ELEMENT DESIGNATOR IS DES-NC-VERIFY  
 PRESENCE IS PRESENCE-OPTIONAL  
 ELEMENT DEFINITION IS  
 'DATA ELEMENTS OF THE LOCATION/ACCESS/RELATIONSHIP'  
 DATA-SUBJECT IS 'SECURITY, GENERAL'  
 USER IS 'NIC-44'  
 RESPONSIBLE FOR DEFINITION  
 SUBORDINATE ELEMENTS ARE  
 NAME-INSPECTOR  
 FACILITY-TYPE  
 STORAGE-TYPE  
 DATE-LAST-ACCRED  
 DATE-INTR-ACCRED  
 DATE-FINL-ACCRED  
 DATE-TSCM-REQ  
 DATE-TSCM-COMP  
 ADD ELEMENT NAME IS CMT-LINE-1204  
 PREPARED BY LET  
 ELEMENT DESCRIPTION IS  
 'LOC ACCESS CMT LINE'  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-UNCLASSIFIED  
 APPLICATION-SYSTEM IS 'SECSS'  
 PICTURE IS X(40)  
 USAGE IS DISPLAY  
 VALUE IS SPACES  
 STORE-VALIDATION IS SSZG  
 MODIFY-VALIDATION IS SMZA  
 ELEMENT DESIGNATOR IS DES-VARIABLE  
 PRESENCE IS PRESENCE-OPTIONAL  
 ELEMENT DEFINITION IS  
 'A LINE OF FREE TEXT COMMENTS'  
 DATA-SUBJECT IS 'SECURITY, GENERAL'  
 USER IS 'NIC-44'  
 RESPONSIBLE FOR DEFINITION  
 ADD ELEMENT NAME IS COMMENT-1204  
 PREPARED BY LET  
 ELEMENT DESCRIPTION IS  
 'LOC ACCESS CMT BLK'  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-UNCLASSIFIED  
 APPLICATION-SYSTEM IS 'SECSS'  
 ELEMENT DESIGNATOR IS DES-VARIABLE  
 PRESENCE IS PRESENCE-OPTIONAL  
 ELEMENT DEFINITION IS  
 'A BLOCK OF 20 COMMENT LINES OF ACCESS'  
 'INSPECTORS COMMENTS'  
 DATA-SUBJECT IS 'SECURITY, GENERAL'  
 USER IS 'NIC-44'  
 RESPONSIBLE FOR DEFINITION  
 SUBORDINATE ELEMENTS ARE  
 CMT-LINE-1204 OCCURS 20  
 ADD ELEMENT NAME IS ID-LOC-ACCESS-CMT  
 PREPARED BY LET  
 ELEMENT DESCRIPTION IS



MEMBER NAME DIESMS  
 \*ACCESS COMMENT IDENTIFIER\*  
 APPLICATION-SYSTEM IS 'SECSS'  
 PRESENCE IS PRESENCE-REQUIRED  
 ELEMENT DEFINITION IS  
 \*THE FIELDS WHICH UNIQUELY IDENTIFY AN\*  
 \*INSPECTORS COMMENTS\*  
 DATA-SUBJECT IS 'SECURITY, GENERAL'  
 USER IS 'NIC-44'  
 RESPONSIBLE FOR DEFINITION  
 SUBORDINATE ELEMENTS ARE  
 DATE-SMS-CMTS  
 SEQ-1204-CMTS.  
 ADD ELEMENT NAME IS DA-LOC-ACCESS-CMT  
 PREPARED BY LET  
 ELEMENT DESCRIPTION IS  
 \*ACCESS COMMENTS\*  
 APPLICATION-SYSTEM IS SECSS  
 PRESENCE IS PRESENCE-OPTIONAL  
 ELEMENT DEFINITION IS  
 \*INSPECTORS COMMENTS ABOUT AN ACCESS\*  
 DATA-SUBJECT IS 'SECURITY, GENERAL'  
 USER IS 'NIC-44'  
 RESPONSIBLE FOR DEFINITION  
 SUBORDINATE ELEMENTS ARE  
 COMMENT-1204.  
 ADD ELEMENT NAME IS ID-BILLET  
 PREPARED BY LET  
 ELEMENT DESCRIPTION IS  
 \*BILLET IDENTIFIER\*  
 APPLICATION-SYSTEM IS 'SECSS'  
 PRESENCE IS PRESENCE-REQUIRED  
 ELEMENT DEFINITION IS  
 \*FIELDS WHICH UNIQUELY IDENTIFY A BILLET\*  
 DATA-SUBJECT IS 'SECURITY, GENERAL'  
 USER IS 'NIC-44'  
 RESPONSIBLE FOR DEFINITION  
 SUBORDINATE ELEMENTS ARE  
 COMMAND-CODE  
 BILLET-NUMBER.  
 ADD ELEMENT NAME IS DA-BILLET  
 PREPARED BY LET  
 ELEMENT DESCRIPTION IS  
 \*BILLET DATA\*  
 APPLICATION-SYSTEM IS 'SECSS'  
 PRESENCE IS PRESENCE-OPTIONAL  
 ELEMENT DEFINITION IS  
 \*THE DATA ASSOCIATED WITH A BILLET\*  
 DATA-SUBJECT IS 'SECURITY, GENERAL'  
 USER IS 'NIC-44'  
 RESPONSIBLE FOR DEFINITION  
 SUBORDINATE ELEMENTS ARE  
 BILLET-SEQ-CODE  
 DATE-BILT-CREATED  
 BILLET-FLAGS  
 TITLE-BILLET.  
 ADD ELEMENT NAME IS ID-ACCESS-TYPE  
 PREPARED BY LET  
 ELEMENT DESCRIPTION IS



MEMBER NAME DIESMS  
 'ACCESS TYPE IDENTIFIER'  
 APPLICATION-SYSTEM IS 'SECS'  
 PRESENCE IS PRESENCE-REQUIRED  
 ELEMENT DEFINITION IS  
 'FIELDS WHICH UNIQUELY IDENTIFY AN ACCESS TYPE'  
 DATA-SUBJECT IS 'SECURITY, GENERAL'  
 USER IS 'NIC-44'  
 RESPONSIBLE FOR DEFINITION  
 SUBORDINATE ELEMENTS ARE  
 ACCESS-CODE.  
 ADD ELEMENT NAME IS DA-ACCESS-TYPE  
 PREPARED BY LET  
 ELEMENT DESCRIPTION IS  
 'ACCESS TYPE DATA'  
 APPLICATION-SYSTEM IS 'SECS'  
 PRESENCE IS PRESENCE-OPTIONAL  
 ELEMENT DEFINITION IS  
 'THE DATA WHICH DEFINES AN ACCESS TYPE'  
 DATA-SUBJECT IS 'SECURITY, GENERAL'  
 USER IS 'NIC-44'  
 RESPONSIBLE FOR DEFINITION  
 SUBORDINATE ELEMENTS ARE  
 ACCESS-FLUAT  
 DATE-ACC-ESTAB  
 DATE-ACC-TERM.  
 ADD ELEMENT NAME IS CITIZENSHIP-TYPE  
 PREPARED BY LET  
 ELEMENT DESCRIPTION IS  
 'TYPE OF CITIZENSHIP'  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-UNCLASSIFIED  
 APPLICATION-SYSTEM IS 'SECS'  
 PICTURE IS X  
 USAGE IS DISPLAY  
 VALUE IS SPACES  
 STORE-VALIDATION IS SSZG  
 MODIFY-VALIDATION IS SMZA  
 ELEMENT DESIGNATOR IS DES-CCCE  
 PRESENCE IS PRESENCE-OPTIONAL  
 ELEMENT DEFINITION IS  
 'THE TYPE OF CITIZENSHIP A PERSON HOLDS'  
 DATA-SUBJECT IS 'SECURITY, GENERAL'  
 USER IS 'NIC-44'  
 RESPONSIBLE FOR DEFINITION  
 RANGE IS 'A'  
 RANGE IS 'B'  
 COMMENTS  
 'A' - NATURAL-BORN CITIZEN  
 'B' - FOREIGN-BORN CITIZEN  
 - ADD ELEMENT NAME IS DATE-CITIZEN  
 PREPARED BY LET  
 SAME AS ELEMENT DATE-COMMCA  
 ELEMENT DESCRIPTION IS  
 'DATE PERSON BECAME CITIZEN'  
 APPLICATION-SYSTEM IS 'SECS'  
 PRESENCE IS PRESENCE-OPTIONAL  
 ELEMENT DEFINITION IS  
 'THE DATE ON WHICH A FOREIGN BORN CITIZEN WAS'

MEMBER\_NAME DISMAS  
 - 'MADE A CITIZEN OF THE COUNTRY'  
 DATA-SUBJECT IS 'SECURITY, GENERAL'  
 USER IS 'NIC-44'  
 RESPONSIBLE FOR DEFINITION.  
 ADD ELEMENT NAME IS DA-CITIZENSHIP  
 PREPARED BY LET  
 ELEMENT DESCRIPTION IS  
 'CITIZENSHIP DATA'  
 APPLICATION-SYSTEM IS SECSS  
 PRESENCE IS PRESENCE-OPTIONAL  
 ELEMENT DEFINITION IS  
 'DATA ON A PERSON CITIZENSHIP'  
 DATA-SUBJECT IS 'SECURITY, GENERAL'  
 USER IS 'NIC-44'  
 RESPONSIBLE FOR DEFINITION  
 SUBORDINATE ELEMENTS ARE  
 CITIZENSHIP-TYPE  
 DATE-CITIZEN.  
 ADD ELEMENT NAME IS DA-BILLET-FECFLE  
 PREPARED BY LET  
 ELEMENT DESCRIPTION IS  
 'BILLET PERSON DATA'  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-UNCLASSIFIED  
 APPLICATION-SYSTEM IS 'SECSS'  
 ELEMENT DESIGNATOR IS DES-NC-VERIFY  
 PRESENCE IS PRESENCE-OPTIONAL  
 ELEMENT DEFINITION IS  
 'DATA ASSOCIATING BILLETS AND PEOPLE'  
 DATA-SUBJECT IS 'SECURITY, GENERAL'  
 USER IS 'NIC-44'  
 RESPONSIBLE FOR DEFINITION  
 SUBORDINATE ELEMENTS ARE  
 DATE-READ-IN  
 DATE-READ-OUT.  
 ADD ELEMENT NAME IS ID-BADGE  
 PREPARED BY LET  
 ELEMENT DESCRIPTION IS  
 'BADGE IDENTIFIER FIELDS'  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-UNCLASSIFIED  
 APPLICATION-SYSTEM IS 'SECSS'  
 ELEMENT DESIGNATOR IS DES-KEY  
 PRESENCE IS PRESENCE-OPTIONAL  
 ELEMENT DEFINITION IS  
 'FIELDS WHICH UNIQUELY IDENTIFY A BADGE'  
 DATA-SUBJECT IS 'SECURITY, GENERAL'  
 USER IS 'NIC-44'  
 RESPONSIBLE FOR DEFINITION  
 SUBORDINATE ELEMENTS ARE  
 BADGE-TYPE-CODE  
 BADGE-NUMBER.  
 ADD ELEMENT NAME IS DA-BADGE  
 PREPARED BY LET  
 ELEMENT DESCRIPTION IS  
 'BADGE DATA ELEMENTS'  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-UNCLASSIFIED

MEMBER NAME DUEMC  
 APPLICATION-SYSTEM IS 'SECSS'  
 ELEMENT DESIGNATOR IS DES-NC-VERIFY  
 PRESENCE IS PRESENCE-OPTIONAL  
 ELEMENT DEFINITION IS  
 'THE AGGREGATE DATA INFORMATION ABOUT A BADGE'  
 DATA-SUBJECT IS 'SECURITY, GENERAL'  
 USER IS 'NIC-44'  
 RESPONSIBLE FOR DEFINITION  
 SUBORDINATE ELEMENTS ARE  
 DATE-BADGE-ISSUE  
 BADGE-DISP-INFO.  
 ADD ELEMENT NAME IS DA-ACCESS-ASSIGNED  
 PREPARED BY LET  
 ELEMENT DESCRIPTION IS  
 'PERSONS ACCESS INFO'  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-UNCLASSIFIED  
 APPLICATION-SYSTEM IS 'SECSS'  
 ELEMENT DESIGNATOR IS DES-NC-VERIFY  
 PRESENCE IS PRESENCE-OPTIONAL  
 ELEMENT DEFINITION IS  
 'THE AGGREGATE DATA ELEMENTS DEFINING A PERSONS'  
 - 'RELATIONSHIP TO AN SECURITY-ACCESS'  
 DATA-SUBJECT IS 'SECURITY, GENERAL'  
 USER IS 'NIC-44'  
 RESPONSIBLE FOR DEFINITION  
 SUBORDINATE ELEMENTS ARE  
 FLG-ACCESS-FLG  
 DATE-ASSIGN-ESTAB  
 DATE-ASSIGN-END  
 DATE-CERT-EXPIRE.  
 ADD ELEMENT NAME IS PEOPLE-VISIT-FLG  
 PREPARED BY 'SDC-1SI'  
 ELEMENT DESCRIPTION IS  
 'VISITOR/VISITES'  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-UNCLASSIFIED  
 APPLICATION-SYSTEM IS 'SECSS'  
 PICTURE IS 9  
 USAGE IS DISPLAY  
 VALUE IS ZEROS  
 STORE-VALIDATION IS SSZN  
 MODIFY-VALIDATION IS SMZV  
 PRESENCE IS PRESENCE-REQUIRED  
 ELEMENT DEFINITION IS  
 'THE NAME OF THE VISITOR OF THE NAME OF THE PERSON  
 - 'BEING VISITED'  
 USER IS 'NIC-44'  
 RESPONSIBLE FOR DEFINITION  
 RANGE IS '1'  
 RANGE IS '2'  
 COMMENTS  
 'THE VALUES ARE  
 - 'A 1 MEANS THE NAME OF THE VISITOR  
 - 'A 2 MEANS THE NAME OF THE PERSON BEING VISITED'.  
 ADD ELEMENT NAME IS DA-PEOPLE-VISIT  
 PREPARED BY 'SDC-1SI'  
 ELEMENT DESCRIPTION IS

MEMBER NAME DIESMS  
 \*NAME OF VISITOR OR PERSON VISITED\*  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-UNCLASSIFIED  
 APPLICATION-SYSTEM IS SECSS  
 PRESENCE IS PRESENCE-RECLIFED  
 ELEMENT DEFINITION IS  
 - \*THE PEOPLE-VISIT RECORD IS THE RELATIONAL RECORD  
 - \*BETWEEN THE PEOPLE AND VISITOR-LOG RECORDS\*  
 DATA SUBJECT IS 'SECURITY, GENERAL'  
 USER IS 'NIC-44'  
 RESPONSIBLE FOR DEFINITION  
 SUBORDINATE ELEMENTS ARE  
 PEOPLE-VISIT-FLG.  
 ADD ELEMENT NAME IS ID-PEOPLE-PERSONAL  
 PREPARED BY 'SOC-ISI'  
 ELEMENT DESCRIPTION IS 'PERSONNEL INFORMATION'  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-UNCLASSIFIED  
 APPLICATION-SYSTEM IS 'SECSS'  
 PRESENCE IS PRESENCE-OPTIONAL  
 ELEMENT DEFINITION IS  
 \*THE UNIQUE CASE CONTROL NUMBER ASSIGNED TO A PERSON\*  
 DATA-SUBJECT IS 'SECURITY GENERAL'  
 USER IS 'NIC-44'  
 RESPONSIBLE FOR DEFINITION  
 SUBORDINATE ELEMENTS ARE  
 BI-CASE-CONTROL.  
 ADD ELEMENT NAME IS DA-PEOPLE-PERSONAL  
 PREPARED BY 'SOC-ISI'  
 ELEMENT DESCRIPTION IS 'PERSONNEL INFORMATION'  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-UNCLASSIFIED  
 APPLICATION-SYSTEM IS 'SECSS'  
 PRESENCE IS PRESENCE-OPTIONAL  
 ELEMENT DEFINITION IS  
 - \*THE PEOPLE-PERSONAL RECORD CONTAINS INFORMATION ABOUT  
 - \*PEOPLE. THE INFORMATION CONTAINED INCLUDES THE  
 - \*RANK OR GRADE, BRANCH OF SERVICE, AND VARIOUS  
 - \*DATES. THERE IS ONE RECORD FOR EACH PERSON. THIS  
 - \*RECORD LINKS TO THE PEOPLE RECORD.\*  
 DATA SUBJECT IS 'SECURITY GENERAL'  
 USER IS 'NIC-44'  
 RESPONSIBLE FOR DEFINITION  
 SUBORDINATE ELEMENTS ARE  
 RANK-SERVICE  
 DATE-BI-REQSTD  
 DATE-BI-STARTED  
 DATE-BI-COMP  
 BI-PENDING-IND.  
 ADD ELEMENT NAME IS URGAN-VISIT-FLG  
 PREPARED BY 'SOC-ISI'  
 ELEMENT DESCRIPTION IS  
 \*ORGANIZATION OF VISITOR/VISITEE\*  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-UNCLASSIFIED  
 APPLICATION-SYSTEM IS 'SECSS'  
 PICTURE IS 9  
 USAGE IS DISPLAY



MEMBER NAME DIESMS  
 VALUE IS ZEROS  
 STOCK-VALIDATION IS SGZN  
 MODIFY-VALIDATION IS SMZV  
 PRESENCE IS PRESENCE-REQUIRED  
 ELEMENT DEFINITION IS  
 - 'THE ORGANIZATION OF THE VISITOR OR THE ORGANIZATION  
 - 'THAT THE VISITOR IS VISITING'  
 USER IS 'NIC-44'  
 RESPONSIBLE FOR DEFINITION  
 RANGE IS '1'  
 RANGE IS '2'  
 COMMENTS  
 - 'THE VALUES ARE  
 - 'A 1 MEANS ORGANIZATION OF VISITOR  
 - 'A 2 MEANS ORGANIZATION VISITED'.  
 ADD ELEMENT NAME IS DA-ORGAN-VISIT  
 PREPARED BY 'SDC-ISI'  
 ELEMENT DESCRIPTION IS  
 'ORGANIZATION OF VISITOR OR ORGANIZATION VISITED'  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-UNCLASSIFIED  
 APPLICATION-SYSTEM IS SECSS  
 PRESENCE IS PRESENCE-REQUIRED  
 ELEMENT DEFINITION IS  
 - 'THE ORGAN-VISIT RECORD IS THE RELATIONAL RECORD  
 - 'BETWEEN THE ORGANIZATION AND VISITOR-LOG RECORDS'.  
 DATA-SUBJECT IS 'SECURITY, GENERAL'  
 USER IS 'NIC-44'  
 RESPONSIBLE FOR DEFINITION  
 SUBORDINATE ELEMENTS ARE  
 ORGAN-VISIT-FLG.  
 ADD ELEMENT NAME IS ID-VISITOR-LOG  
 PREPARED BY SDC-ISI  
 ELEMENT DESCRIPTION IS 'VISITOR LOG'  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-UNCLASSIFIED  
 APPLICATION-SYSTEM IS 'SECSS'  
 PRESENCE IS PRESENCE-OPTIONAL  
 ELEMENT DEFINITION IS  
 'THIS FIELD IS FOR THE LOG IN DATE AND TIME'  
 DATA-SUBJECT IS 'SECURITY, GENERAL'  
 USER IS 'NIC-44'  
 RESPONSIBLE FOR DEFINITION  
 SUBORDINATE ELEMENTS ARE  
 DTG-LOG-IN.  
 ADD ELEMENT NAME IS DA-VISITOR-LOG  
 PREPARED BY SDC-ISI  
 ELEMENT DESCRIPTION IS 'VISITOR LOG'  
 ENTRY-SECURITY IS ENTRY-UNCLASSIFIED  
 DATA-SECURITY IS DATA-UNCLASSIFIED  
 APPLICATION-SYSTEM IS 'SECSS'  
 PRESENCE IS PRESENCE-OPTIONAL  
 ELEMENT DEFINITION IS  
 - 'THE VISITOR LOG RECORD CONTAINS INFORMATION ABOUT  
 - 'VISITORS TO A FACILITY. THE INFORMATION CONTAINED  
 - 'INCLUDES THE NAME OF THE VISITOR, THE NAME OF THE  
 - 'POINT OF CONTACT, THE PROFESSIONAL AFFILIATION OF THE  
 - 'VISITOR, AND DATES AND TIMES OF ENTRY AND EXIT. THERE



MEMBER NAME DIESMS

- IS ONE RECORD FOR EACH PERSON. THIS RECORD LINKS TO

- THE PEOPLE AND LOCATION RECORDS.

DATA-SUBJECT IS 'SECURITY, GENERAL'

USER IS 'NIC-44'

RESPONSIBLE FOR DEFINITION

SUBORDINATE ELEMENTS ARE

DTG-LOG-OUT

PURPOSE-OF-VISIT.

```

01-01 NOSEQ
01-01 SCHEMA DESCRIPTION.
01-01 SCHEMA NAME IS UDB VERSION 0101.
01-01 AUTHOR. J.E. TOWNER.
01-01 DATE. 02/09/79.
01-01 INSTALLATION. NIPSSA.
01-01
01-01 FILE DESCRIPTION.
01-01
01-01
01-01 FILE NAME IS OLQ2FILE ASSIGN TO OLQ2FILE.
01-01
01-01 FILE NAME IS DATAVAL ASSIGN TO DATAVAL.
01-01
01-01 FILE NAME IS UDBINDEX ASSIGN TO UDBINDEX.
01-01
01-01 FILE NAME IS UDBFIL1 ASSIGN TO UDBFIL1.
01-01 FILE NAME IS UDBFIL2 ASSIGN TO UDBFIL2.
01-01 FILE NAME IS UDBFIL3 ASSIGN TO UDBFIL3.
01-01 FILE NAME IS UDBFIL4 ASSIGN TO UDBFIL4.
01-01 FILE NAME IS UDBFIL5 ASSIGN TO UDBFIL5.
01-01 FILE NAME IS UDBFIL6 ASSIGN TO UDBFIL6.
01-01 FILE NAME IS UDBFIL7 ASSIGN TO UDBFIL7.
01-01 FILE NAME IS UDBFIL8 ASSIGN TO UDBFIL8.
01-01 FILE NAME IS UDBFIL9 ASSIGN TO UDBFIL9.
01-01 FILE NAME IS UDBFIL10 ASSIGN TO UDBFIL10.
01-01 FILE NAME IS UDBFIL11 ASSIGN TO UDBFIL11.
01-01 FILE NAME IS UDBFIL12 ASSIGN TO UDBFIL12.
01-01 FILE NAME IS UDBFIL13 ASSIGN TO UDBFIL13.
01-01 FILE NAME IS UDBFIL14 ASSIGN TO UDBFIL14.
01-01 FILE NAME IS UDBFIL15 ASSIGN TO UDBFIL15.
01-01
01-01 AREA DESCRIPTION.
01-01
01-01
01-01 *      1 PAGE/TRACK = 7292 BYTES
01-01 *      2 PAGES/TRACK = 3520 BYTES
01-01 *      3 PAGES/TRACK = 2296 BYTES
01-01 *      4 PAGES/TRACK = 1690 BYTES
01-01 *      5 PAGES/TRACK = 1332 BYTES
01-01
01-01 AREA NAME IS VALIDATION
01-01 RANGE IS 500001 THRU 500500
01-01 WITHIN FILE DATAVAL FROM 1 THRU 500.
01-01
01-01 AREA NAME IS DDLCRUN
01-01 RANGE IS 600001 THRU 600150
01-01 WITHIN FILE OLQ2FILE FROM 1 THRU 150.
01-01
01-01 AREA NAME IS UDB-INDEX-AREA
01-01 RANGE IS 6000001 THRU 6000316
01-01 WITHIN FILE UDBINDEX FROM 1 THRU 316.
01-01
01-01 AREA NAME IS UDBAREA
01-01 RANGE IS 7000001 THRU 7059550
01-01 WITHIN FILE UDBFIL1 FROM 1 THRU 3970.
01-01 WITHIN FILE UDBFIL2 FROM 1 THRU 3970.
01-01 WITHIN FILE UDBFIL3 FROM 1 THRU 3970.
01-01 WITHIN FILE UDBFIL4 FROM 1 THRU 3970.

```

01-01	WITHIN FILE UDBFIL5 FROM 1 THRU 3970;
01-01	WITHIN FILE UDBFIL6 FROM 1 THRU 3970;
01-01	WITHIN FILE UDBFIL7 FROM 1 THRU 3970;
01-01	WITHIN FILE UDBFIL8 FROM 1 THRU 3970;
01-01	WITHIN FILE UDBFIL9 FROM 1 THRU 3970;
01-01	WITHIN FILE UDBFIL10 FROM 1 THRU 3970;
01-01	WITHIN FILE UDBFIL11 FROM 1 THRU 3970;
01-01	WITHIN FILE UDBFIL12 FROM 1 THRU 3970;
01-01	WITHIN FILE UDBFIL13 FROM 1 THRU 3970;
01-01	WITHIN FILE UDBFIL14 FROM 1 THRU 3970;
01-01	WITHIN FILE UDBFIL15 FROM 1 THRU 3970.
01-01	
01-01	RECORD DESCRIPTION.
01-01	
01-01	
01-01	
01-01	
01-01	RECORD NAME IS DATA-VALID.
01-01	RECORD ID IS 5199.
01-01	LOCATION MODE IS CALC USING VALID-PARM DUPLICATES NOT.
01-01	WITHIN VALIDATION AREA.
01-01	
01-01	03 VALID-PARM.
01-01	05 VP-DATAELE PIC X(16).
01-01	*DATA ELEMENT NAME OF ELEMENT WITHIN DB RCD*.
01-01	05 VP-VALUE PIC X(20).
01-01	*VALID DATA VALUE FOR ELEMENT NAMED*.
01-01	
01-01	
001000	RECORD NAME IS IXSUB.
001100	RECORD ID IS 9995.
001200	LOCATION MODE IS CALC
001300	USING IXSSN
	DUPLICATES ARE NOT ALLOWED.
	WITHIN UDB-INDEX-AREA AREA
	FROM 6000002 THRU 6000002.
001600	
001700	03 IXSSN
001800	PIC X(8).
001900	
002000	RECORD NAME IS IXREC.
002100	RECORD ID IS 9996.
002200	LOCATION MODE IS VIA IXSUB-IXREC SET.
	WITHIN UDB-INDEX-AREA AREA
	FROM 6000002 THRU 6000002.
002500	
002600	03 ISKDSP
002700	PIC S9(4) COMP.
002800	03 FILLER
002900	PIC X(6).
003000	
003100	RECORD NAME IS IXSET.
003200	RECORD ID IS 9997.
003300	LOCATION MODE IS CALC
	USING ISNAME
	DUPLICATES ARE NOT ALLOWED.
	WITHIN UDB-INDEX-AREA AREA
	FROM 6000002 THRU 6000002.
003600	
003700	03 ISVIA
003800	PIC S9(4) COMP.
003900	03 ISTER
	PIC S9(4) COMP.
004000	03 ISTET
	PIC S9(4) COMP.
004100	03 ISCFST
	PIC S9(4) COMP.
004200	03 ISTO
	PIC S9(4) COMP.
	03 ISKL
	PIC S9(4) COMP.

004300	03	ISDBKL	PIC S9(4) COMP.
004400	03	ISMBRO	PIC S9(4) COMP.
004500	03	ISOPT	PIC S9(4) COMP.
004600	03	ISEQ	PIC S9(4) COMP.
004700	03	ISRECN	PIC X(16).
004800	03	ISNAME	PIC X(16).
004900			
005000			
005100		RECORD NAME IS IXDET.	
005200		RECORD ID IS 9998.	
005300		LOCATION MODE IS DIRECT.	
		WITHIN UDB-INDEX-AREA AREA	
		FROM 6000004 THRU 6000316.	
005600		MINIMUM ROOT LENGTH IS 12 CHARACTERS.	
005700		MINIMUM FRAGMENT LENGTH IS 8 CHARACTERS.	
005800			
005900	03	IDODBK	PIC S9(8) COMP.
006000	03	IDTOP	PIC S9(4) COMP.
006100	03	IDBOT	PIC S9(4) COMP.
006200	03	IDO	PIC S9(4) COMP.
006300	03	IDL	PIC S9(4) COMP.
006400	03	IDTBL	
006500	05	IDENTRY	PIC X
006600			OCCURS 0 TO 788 TIMES
006700			DEPENDING ON IDL.
006800			
006900			
007000		RECORD NAME IS IXOWNER.	
007100		RECORD ID IS 9999.	
007200		LOCATION MODE IS DIRECT.	
		WITHIN UDB-INDEX-AREA AREA.	
007400			
007500	03	FILLER	PIC X(8).
007600		COMMENT 'THIS IS A DUMMY RECORD THAT WILL NOT OCCUR	
007700-		'IN THE DATABASE'.	
007800			
007900*			
01-01			
01-01		RECORD NAME IS UDB-HDR.	
01-01		RECORD ID IS 5411.	
01-01		LOCATION MODE IS CALC USING UDB-IDENT DUPLICATES NOT.	
01-01		WITHIN UDBAREA AREA.	
01-01		MINIMUM ROOT LENGTH IS CONTROL.	
01-01		MINIMUM FRAGMENT LENGTH IS 24.	
01-01		CALL IDMSCOMP BEFORE STORE.	
01-01		CALL IDMSCOMP BEFORE MODIFY.	
01-01		CALL IDMSDCOM AFTER GET.	
01-01		CALL IDMSSTRC BEFORE STORE.	
01-01		CALL IDMSMODC BEFORE MODIFY.	
01-01		CALL IDMSDELIC BEFORE ERASE.	
01-01		CALL IDMSACCC AFTER GET.	
01-01	03	CTL-UDB-HDR	PIC S9(8) COMP VALUE +0.
01-01	03	RCD-UDBHUR.	
01-01	05	UDB-IDENT	PIC X(8) VALUE SPACES.
01-01	05	DATE-CREATED-UDB	PIC 9(6) VALUE 0.
01-01	05	DATE-UPDATED-UDB	PIC 9(6) VALUE 0.
01-01	05	UDB-ENTRY-CT	PIC S9(8) COMP VALUE +0.
01-01	05	UDB-ENTRY-LIMIT	PIC S9(8) COMP VALUE +0.
01-01	05	UDB-UPDATE-CT	PIC S9(8) COMP VALUE +0.
01-01	05	UDB-RELATE-CT	PIC S9(8) COMP VALUE +0.
01-01	05	UDB-CLASS	PIC X VALUE SPACE.
01-01	05	UDB-HANDL	PIC XX VALUE SPACES.
01-01	05	UDB-TITLE	PIC X(40) VALUE SPACES.



	05	UDB-IPID	PIC XXXX	VALUE SPACES.
01-01				
01-01				
01-01				
01-01		RECORD NAME IS ORGANIZATION.		
01-01		RECORD ID IS 1000.		
01-01		LOCATION MODE IS CALC USING ID-ORGANIZATION	DUPLICATES NOT.	
01-01		WITHIN UDBAREA AREA.		
01-01		MINIMUM ROOT LENGTH IS CONTROL.		
01-01		MINIMUM FRAGMENT LENGTH IS 24.		
01-01		CALL IDMSCOMP BEFORE STORE.		
01-01		CALL IDMSCOMP BEFORE MODIFY.		
01-01		CALL IDMSUCOM AFTER GET.		
01-01		COPY ORGANIZATION	RECORD.	
01-01				
01-01		RECORD NAME IS REFERENCE.		
01-01		RECORD ID IS 1002.		
01-01		LOCATION MODE IS CALC USING ID-REFERENCE	DUPLICATES NOT.	
01-01		WITHIN UDBAREA AREA.		
01-01		MINIMUM ROOT LENGTH IS CONTROL.		
01-01		MINIMUM FRAGMENT LENGTH IS 24.		
01-01		CALL IDMSCOMP BEFORE STORE.		
01-01		CALL IDMSCOMP BEFORE MODIFY.		
01-01		CALL IDMSUCOM AFTER GET.		
01-01		COPY REFERENCE	RECORD.	
01-01				
01-01		RECORD NAME IS PEOPLE.		
01-01		RECORD ID IS 1003.		
01-01		LOCATION MODE IS CALC USING ID-PEOPLE	DUPLICATES NOT.	
01-01		WITHIN UDBAREA AREA.		
01-01		MINIMUM ROOT LENGTH IS CONTROL.		
01-01		MINIMUM FRAGMENT LENGTH IS 24.		
01-01		CALL IDMSCOMP BEFORE STORE.		
01-01		CALL IDMSCOMP BEFORE MODIFY.		
01-01		CALL IDMSUCOM AFTER GET.		
01-01		COPY PEOPLE	RECORD.	
01-01				
01-01		RECORD NAME IS NIM-PROJECT.		
01-01		RECORD ID IS 1004.		
01-01		LOCATION MODE IS CALC USING ID-NIM-PROJECT	DUPLICATES NOT.	
01-01		WITHIN UDBAREA AREA.		
01-01		MINIMUM ROOT LENGTH IS CONTROL.		
01-01		MINIMUM FRAGMENT LENGTH IS 24.		
01-01		CALL IDMSCOMP BEFORE STORE.		
01-01		CALL IDMSCOMP BEFORE MODIFY.		
01-01		CALL IDMSUCOM AFTER GET.		
01-01		COPY NIM-PROJECT	RECORD.	
01-01				
01-01		RECORD NAME IS ORGAN-COMMENTS.		
01-01		RECORD ID IS 1005.		
01-01		LOCATION MODE IS VIA NIMIS-1019-A.		
01-01		WITHIN UDBAREA AREA.		
01-01		MINIMUM ROOT LENGTH IS CONTROL.		
01-01		MINIMUM FRAGMENT LENGTH IS 24.		
01-01		CALL IDMSCOMP BEFORE STORE.		
01-01		CALL IDMSCOMP BEFORE MODIFY.		
01-01		CALL IDMSUCOM AFTER GET.		
01-01		COPY ORGAN-COMMENTS	RECORD.	
01-01				
01-01		RECORD NAME IS POSIT-DESCR.		
01-01		RECORD ID IS 1006.		
01-01		LOCATION MODE IS VIA NIMIS-1019-A.		
01-01		WITHIN UDBAREA AREA.		
01-01		MINIMUM ROOT LENGTH IS CONTROL.		



```

01-01 MINIMUM FRAGMENT LENGTH IS 24.
01-01 CALL IDMSCOMP BEFORE STORE.
01-01 CALL IDMSCOMP BEFORE MODIFY.
01-01 CALL IDMSUCOM AFTER GET.
01-01 COPY POSIT-DESCR RECORD.
01-01
01-01 RECORD NAME IS DUM-POSIT-DESCR.
01-01 RECORD ID IS 1007.
01-01 LOCATION MODE IS VIA NIMIS-1006-A.
01-01 WITHIN UDBAREA AREA.
01-01 03 ID-DUM-POSIT-DESCR PIC xxxx.
01-01
01-01 RECORD NAME IS NIM-COMMENTS.
01-01 RECORD ID IS 1008.
01-01 LOCATION MODE IS VIA NIMIS-1004-A.
01-01 WITHIN UDBAREA AREA.
01-01 MINIMUM ROOT LENGTH IS CONTROL.
01-01 MINIMUM FRAGMENT LENGTH IS 24.
01-01 CALL IDMSCOMP BEFORE STORE.
01-01 CALL IDMSCOMP BEFORE MODIFY.
01-01 CALL IDMSUCOM AFTER GET.
01-01 COPY NIM-COMMENTS RECORD.
01-01
01-01 RECORD NAME IS NIM-MSTONES.
01-01 RECORD ID IS 1009.
01-01 LOCATION MODE IS VIA NIMIS-1004-A.
01-01 WITHIN UDBAREA AREA.
01-01 MINIMUM ROOT LENGTH IS CONTROL.
01-01 MINIMUM FRAGMENT LENGTH IS 24.
01-01 CALL IDMSCOMP BEFORE STORE.
01-01 CALL IDMSCOMP BEFORE MODIFY.
01-01 CALL IDMSUCOM AFTER GET.
01-01 COPY NIM-MSTONES RECORD.
01-01
01-01 RECORD NAME IS DUM-NIM-MSTONES.
01-01 RECORD ID IS 1010.
01-01 LOCATION MODE IS VIA NIMIS-1009-A.
01-01 WITHIN UDBAREA AREA.
01-01 03 ID-DUM-NIM-MSTONES PIC xxxx.
01-01
01-01 RECORD NAME IS NIM-SUBTASK.
01-01 RECORD ID IS 1011.
01-01 LOCATION MODE IS VIA NIMIS-1004-A.
01-01 WITHIN UDBAREA AREA.
01-01 MINIMUM ROOT LENGTH IS CONTROL.
01-01 MINIMUM FRAGMENT LENGTH IS 24.
01-01 CALL IDMSCOMP BEFORE STORE.
01-01 CALL IDMSCOMP BEFORE MODIFY.
01-01 CALL IDMSUCOM AFTER GET.
01-01 COPY NIM-SUBTASK RECORD.
01-01
01-01 RECORD NAME IS DUM-NIM-SUBTASK.
01-01 RECORD ID IS 1012.
01-01 LOCATION MODE IS VIA NIMIS-1011-A.
01-01 WITHIN UDBAREA AREA.
01-01 03 ID-DUM-NIM-SUBTASK PIC xxxx.
01-01
01-01 RECORD NAME IS NIM-SUBTASK-CMTS.
01-01 RECORD ID IS 1013.
01-01 LOCATION MODE IS VIA NIMIS-1011-A.
01-01 WITHIN UDBAREA AREA.
01-01 MINIMUM ROOT LENGTH IS CONTROL.
01-01 MINIMUM FRAGMENT LENGTH IS 24.

```

01-01 CALL IDMSCOMP BEFORE STORE.  
 01-01 CALL IDMSCOMP BEFORE MODIFY.  
 01-01 CALL IDMSDCOM AFTER GET.  
 01-01 COPY NIM-SUBTASK-CMTS RECORD.  
 01-01  
 01-01 RECORD NAME IS NIM-MSTONE-CMTS.  
 01-01 RECORD ID IS 1014.  
 01-01 LOCATION MODE IS VIA NIMIS-1009-A.  
 01-01 WITHIN UDBAREA AREA.  
 01-01 MINIMUM ROOT LENGTH IS CONTROL.  
 01-01 MINIMUM FRAGMENT LENGTH IS 24.  
 01-01 CALL IDMSCOMP BEFORE STORE.  
 01-01 CALL IDMSCOMP BEFORE MODIFY.  
 01-01 CALL IDMSDCOM AFTER GET.  
 01-01 COPY NIM-MSTONE-CMTS RECORD.  
 01-01  
 01-01 RECORD NAME IS NIM-LABORHRS.  
 01-01 RECORD ID IS 1015.  
 01-01 LOCATION MODE IS VIA NIMIS-1004-A.  
 01-01 WITHIN UDBAREA AREA.  
 01-01 MINIMUM ROOT LENGTH IS CONTROL.  
 01-01 MINIMUM FRAGMENT LENGTH IS 24.  
 01-01 CALL IDMSCOMP BEFORE STORE.  
 01-01 CALL IDMSCOMP BEFORE MODIFY.  
 01-01 CALL IDMSDCOM AFTER GET.  
 01-01 COPY NIM-LABORHRS RECORD.  
 01-01  
 01-01 RECORD NAME IS ORG-NIMPROJ.  
 01-01 RECORD ID IS 1016.  
 01-01 LOCATION MODE IS VIA NIMIS-1019-R.  
 01-01 WITHIN UDBAREA AREA.  
 01-01 MINIMUM ROOT LENGTH IS CONTROL.  
 01-01 MINIMUM FRAGMENT LENGTH IS 24.  
 01-01 CALL IDMSCOMP BEFORE STORE.  
 01-01 CALL IDMSCOMP BEFORE MODIFY.  
 01-01 CALL IDMSDCOM AFTER GET.  
 01-01 COPY ORG-NIMPROJ RECORD.  
 01-01  
 01-01 RECORD NAME IS NIMPROJ-RELATE.  
 01-01 RECORD ID IS 1017.  
 01-01 LOCATION MODE IS VIA NIMIS-1004-A.  
 01-01 WITHIN UDBAREA AREA.  
 01-01 MINIMUM ROOT LENGTH IS CONTROL.  
 01-01 MINIMUM FRAGMENT LENGTH IS 24.  
 01-01 CALL IDMSCOMP BEFORE STORE.  
 01-01 CALL IDMSCOMP BEFORE MODIFY.  
 01-01 CALL IDMSDCOM AFTER GET.  
 01-01 COPY NIMPROJ-RELATE RECORD.  
 01-01  
 01-01 RECORD NAME IS PEOPLE-NIMIS.  
 01-01 RECORD ID IS 1018.  
 01-01 LOCATION MODE IS VIA UDB-1003-A.  
 01-01 WITHIN UDBAREA AREA.  
 01-01 MINIMUM ROOT LENGTH IS CONTROL.  
 01-01 MINIMUM FRAGMENT LENGTH IS 24.  
 01-01 CALL IDMSCOMP BEFORE STORE.  
 01-01 CALL IDMSCOMP BEFORE MODIFY.  
 01-01 CALL IDMSDCOM AFTER GET.  
 01-01 COPY PEOPLE-NIMIS RECORD.  
 01-01  
 01-01 RECORD NAME IS ORG-NIMIS.  
 01-01 RECORD ID IS 1019.  
 01-01 LOCATION MODE IS VIA UDB-1000-A.

01-01	WITHIN UDBAREA AREA.
01-01	MINIMUM ROOT LENGTH IS CONTROL.
01-01	MINIMUM FRAGMENT LENGTH IS 24.
01-01	CALL IDMSCOMP BEFORE STORE.
01-01	CALL IDMSCOMP BEFORE MODIFY.
01-01	CALL IDMSUCOM AFTER GET.
01-01	COPY ORG-NIMIS RECORD.
01-01	RECORD NAME IS BILLET.
01-01	RECORD ID IS 1200.
01-01	LOCATION MODE IS CALC USING ID-BILLET DUPLICATES NOT.
01-01	WITHIN UDBAREA AREA.
01-01	MINIMUM ROOT LENGTH IS CONTROL.
01-01	MINIMUM FRAGMENT LENGTH IS 24.
01-01	CALL IDMSCOMP BEFORE STORE.
01-01	CALL IDMSCOMP BEFORE MODIFY.
01-01	CALL IDMSUCOM AFTER GET.
01-01	COPY BILLET RECORD.
01-01	RECORD NAME IS ACCESS-TYPE.
01-01	RECORD ID IS 1201.
01-01	LOCATION MODE IS CALC USING ID-ACCESS-TYPE DUPLICATES NOT.
01-01	WITHIN UDBAREA AREA.
01-01	MINIMUM ROOT LENGTH IS CONTROL.
01-01	MINIMUM FRAGMENT LENGTH IS 24.
01-01	CALL IDMSCOMP BEFORE STORE.
01-01	CALL IDMSCOMP BEFORE MODIFY.
01-01	CALL IDMSUCOM AFTER GET.
01-01	COPY ACCESS-TYPE RECORD.
01-01	RECORD NAME IS UDB-LOCATION.
01-01	RECORD ID IS 1202.
01-01	LOCATION MODE IS CALC USING ID-UDB-LOCATION DUPLICATES NOT.
01-01	WITHIN UDBAREA AREA.
01-01	MINIMUM ROOT LENGTH IS CONTROL.
01-01	MINIMUM FRAGMENT LENGTH IS 24.
01-01	CALL IDMSCOMP BEFORE STORE.
01-01	CALL IDMSCOMP BEFORE MODIFY.
01-01	CALL IDMSUCOM AFTER GET.
01-01	COPY UDB-LOCATION RECORD.
01-01	RECORD NAME IS LOCATION-ACCESS.
01-01	RECORD ID IS 1203.
01-01	LOCATION MODE IS VIA UDB-1202-A.
01-01	WITHIN UDBAREA AREA.
01-01	MINIMUM ROOT LENGTH IS CONTROL.
01-01	MINIMUM FRAGMENT LENGTH IS 24.
01-01	CALL IDMSCOMP BEFORE STORE.
01-01	CALL IDMSCOMP BEFORE MODIFY.
01-01	CALL IDMSUCOM AFTER GET.
01-01	COPY LOCATION-ACCESS RECORD.
01-01	RECORD NAME IS LOC-ACCESS-CMT.
01-01	RECORD ID IS 1204.
01-01	LOCATION MODE IS VIA SMS-1203-A.
01-01	WITHIN UDBAREA AREA.
01-01	MINIMUM ROOT LENGTH IS CONTROL.

```

01-01 MINIMUM FRAGMENT LENGTH IS 24.
01-01 CALL IDMSCOMP BEFORE STORE.
01-01 CALL IDMSCOMP BEFORE MODIFY.
01-01 CALL IDMSUCOM AFTER GET.
01-01 COPY LOC-ACCESS-CMT RECORD.
01-01
01-01 RECORD NAME IS LOCATION-RELATE.
01-01 RECORD ID IS 1205.
01-01 LOCATION MODE IS VIA UDB-1202-A.
01-01 WITHIN UDBAREA AREA.
01-01 MINIMUM ROOT LENGTH IS CONTROL.
01-01 MINIMUM FRAGMENT LENGTH IS 24.
01-01 CALL IDMSCOMP BEFORE STORE.
01-01 CALL IDMSCOMP BEFORE MODIFY.
01-01 CALL IDMSUCOM AFTER GET.
01-01 COPY LOCATION-RELATE RECORD.
01-01
01-01 RECORD NAME IS SSO-CONTROL.
01-01 RECORD ID IS 1206.
01-01 LOCATION MODE IS VIA UDB-1000-A.
01-01 WITHIN UDBAREA AREA.
01-01 MINIMUM ROOT LENGTH IS CONTROL.
01-01 MINIMUM FRAGMENT LENGTH IS 24.
01-01 CALL IDMSCOMP BEFORE STORE.
01-01 CALL IDMSCOMP BEFORE MODIFY.
01-01 CALL IDMSUCOM AFTER GET.
01-01 COPY SSO-CONTROL RECORD.
01-01
01-01 RECORD NAME IS LOCATION-ORGAN.
01-01 RECORD ID IS 1207.
01-01 LOCATION MODE IS VIA UDB-1000-B.
01-01 WITHIN UDBAREA AREA.
01-01 MINIMUM ROOT LENGTH IS CONTROL.
01-01 MINIMUM FRAGMENT LENGTH IS 24.
01-01 CALL IDMSCOMP BEFORE STORE.
01-01 CALL IDMSCOMP BEFORE MODIFY.
01-01 CALL IDMSUCOM AFTER GET.
01-01 COPY LOCATION-ORGAN RECORD.
01-01
01-01 RECORD NAME IS ACCESS-TYPE-TYPE.
01-01 RECORD ID IS 1208.
01-01 LOCATION MODE IS VIA SMS-1201-A.
01-01 WITHIN UDBAREA AREA.
01-01 MINIMUM ROOT LENGTH IS CONTROL.
01-01 MINIMUM FRAGMENT LENGTH IS 24.
01-01 CALL IDMSCOMP BEFORE STORE.
01-01 CALL IDMSCOMP BEFORE MODIFY.
01-01 CALL IDMSUCOM AFTER GET.
01-01 COPY ACCESS-TYPE-TYPE RECORD.
01-01
01-01 RECORD NAME IS ORGAN-BILLET.
01-01 RECORD ID IS 1209.
01-01 LOCATION MODE IS VIA SMS-1200-B.
01-01 WITHIN UDBAREA AREA.
01-01 MINIMUM ROOT LENGTH IS CONTROL.
01-01 MINIMUM FRAGMENT LENGTH IS 24.
01-01 CALL IDMSCOMP BEFORE STORE.
01-01 CALL IDMSCOMP BEFORE MODIFY.

```



01-01	CALL IDMSUCOM AFTER GET.
01-01	COPY ORGAN-BILLET RECORD.
01-01	
01-01	RECORD NAME IS ACCESS-BILLET.
01-01	RECORD ID IS 1210.
01-01	LOCATION MODE IS VIA SMS-1200-A.
01-01	WITHIN UDBAREA AREA.
01-01	MINIMUM ROOT LENGTH IS CONTROL.
01-01	MINIMUM FRAGMENT LENGTH IS 24.
01-01	CALL IDMSCOMP BEFORE STORE.
01-01	CALL IDMSCOMP BEFORE MODIFY.
01-01	CALL IDMSUCOM AFTER GET.
01-01	COPY ACCESS-BILLET RECORD.
01-01	
01-01	RECORD NAME IS VISITOR-LOG.
01-01	RECORD ID IS 1211.
01-01	LOCATION MODE IS VIA UDB-1202-A.
01-01	WITHIN UDBAREA AREA.
01-01	MINIMUM ROOT LENGTH IS CONTROL.
01-01	MINIMUM FRAGMENT LENGTH IS 24.
01-01	CALL IDMSCOMP BEFORE STORE.
01-01	CALL IDMSCOMP BEFORE MODIFY.
01-01	CALL IDMSUCOM AFTER GET.
01-01	COPY VISITOR-LOG RECORD.
01-01	
01-01	RECORD NAME IS ORGAN-VISIT.
01-01	RECORD ID IS 1212.
01-01	LOCATION MODE IS VIA SMS-1211-B.
01-01	WITHIN UDBAREA AREA.
01-01	MINIMUM ROOT LENGTH IS CONTROL.
01-01	MINIMUM FRAGMENT LENGTH IS 24.
01-01	CALL IDMSCOMP BEFORE STORE.
01-01	CALL IDMSCOMP BEFORE MODIFY.
01-01	CALL IDMSUCOM AFTER GET.
01-01	COPY ORGAN-VISIT RECORD.
01-01	
01-01	RECORD NAME IS PEOPLE-VISIT.
01-01	RECORD ID IS 1213.
01-01	LOCATION MODE IS VIA SMS-1211-B.
01-01	WITHIN UDBAREA AREA.
01-01	MINIMUM ROOT LENGTH IS CONTROL.
01-01	MINIMUM FRAGMENT LENGTH IS 24.
01-01	CALL IDMSCOMP BEFORE STORE.
01-01	CALL IDMSCOMP BEFORE MODIFY.
01-01	CALL IDMSUCOM AFTER GET.
01-01	COPY PEOPLE-VISIT RECORD.
01-01	
01-01	RECORD NAME IS BADGE.
01-01	RECORD ID IS 1214.
01-01	LOCATION MODE IS VIA UDB-1000-A.
01-01	WITHIN UDBAREA AREA.
01-01	MINIMUM ROOT LENGTH IS CONTROL.
01-01	MINIMUM FRAGMENT LENGTH IS 24.
01-01	CALL IDMSCOMP BEFORE STORE.
01-01	CALL IDMSCOMP BEFORE MODIFY.
01-01	CALL IDMSUCOM AFTER GET.



01-01 COPY BADGE RECORD.  
 01-01  
 01-01  
 01-01 RECORD NAME IS ORGAN-PEOPLE.  
 01-01 RECORD ID IS 1215.  
 01-01 LOCATION MODE IS VIA SMS-1219-B.  
 01-01 WITHIN UDBAREA AREA.  
 01-01 MINIMUM ROOT LENGTH IS CONTROL.  
 01-01 MINIMUM FRAGMENT LENGTH IS 24.  
 01-01 CALL IDMSCOMP BEFORE STORE.  
 01-01 CALL IDMSCOMP BEFORE MODIFY.  
 01-01 CALL IDMSDCOM AFTER GET.  
 01-01 COPY ORGAN-PEOPLE RECORD.  
 01-01  
 01-01  
 01-01 RECORD NAME IS ACCESS-ASSIGNED.  
 01-01 RECORD ID IS 1216.  
 01-01 LOCATION MODE IS VIA SMS-1219-A.  
 01-01 WITHIN UDBAREA AREA.  
 01-01 MINIMUM ROOT LENGTH IS CONTROL.  
 01-01 MINIMUM FRAGMENT LENGTH IS 24.  
 01-01 CALL IDMSCOMP BEFORE STORE.  
 01-01 CALL IDMSCOMP BEFORE MODIFY.  
 01-01 CALL IDMSDCOM AFTER GET.  
 01-01 COPY ACCESS-ASSIGNED RECORD.  
 01-01  
 01-01  
 01-01 RECORD NAME IS CITIZENSHIP.  
 01-01 RECORD ID IS 1217.  
 01-01 LOCATION MODE IS VIA SMS-1219-B.  
 01-01 WITHIN UDBAREA AREA.  
 01-01 MINIMUM ROOT LENGTH IS CONTROL.  
 01-01 MINIMUM FRAGMENT LENGTH IS 24.  
 01-01 CALL IDMSCOMP BEFORE STORE.  
 01-01 CALL IDMSCOMP BEFORE MODIFY.  
 01-01 CALL IDMSDCOM AFTER GET.  
 01-01 COPY CITIZENSHIP RECORD.  
 01-01  
 01-01  
 01-01 RECORD NAME IS BILLET-PEOPLE.  
 01-01 RECORD ID IS 1218.  
 01-01 LOCATION MODE IS VIA SMS-1219-B.  
 01-01 WITHIN UDBAREA AREA.  
 01-01 MINIMUM ROOT LENGTH IS CONTROL.  
 01-01 MINIMUM FRAGMENT LENGTH IS 24.  
 01-01 CALL IDMSCOMP BEFORE STORE.  
 01-01 CALL IDMSCOMP BEFORE MODIFY.  
 01-01 CALL IDMSDCOM AFTER GET.  
 01-01  
 01-01 COPY BILLET-PEOPLE RECORD.  
 01-01  
 01-01  
 01-01 RECORD NAME IS PEOPLE-SMS.  
 01-01 RECORD ID IS 1219.  
 01-01 LOCATION MODE IS VIA U08-1003-A.  
 01-01 WITHIN UDBAREA AREA.  
 01-01 MINIMUM ROOT LENGTH IS CONTROL.  
 01-01 MINIMUM FRAGMENT LENGTH IS 24.  
 01-01 CALL IDMSCOMP BEFORE STORE.  
 01-01 CALL IDMSCOMP BEFORE MODIFY.  
 01-01 CALL IDMSDCOM AFTER GET.  
 01-01 COPY PEOPLE-SMS RECORD.  
 01-01

```

01-01
01-01 RECORD NAME IS PEOPLE-RELATE.
01-01 RECORD ID IS 1220.
01-01 LOCATION MODE IS VIA UDB-1003-A.
01-01 WITHIN UDBAREA AREA.
01-01 MINIMUM ROOT LENGTH IS CONTROL.
01-01 MINIMUM FRAGMENT LENGTH IS 24.
01-01 CALL IDMSCOMP BEFORE STORE.
01-01 CALL IDMSCOMP BEFORE MODIFY.
01-01 CALL IDMSUCOM AFTER GET.
01-01 COPY PEOPLE-RELATE RECORD.
01-01
01-01 RECORD NAME IS BILLET-AA.
01-01 RECORD ID IS 1221.
01-01 LOCATION MODE IS VIA SMS-1200-B.
01-01 WITHIN UDBAREA AREA.
01-01 MINIMUM ROOT LENGTH IS CONTROL.
01-01 MINIMUM FRAGMENT LENGTH IS 24.
01-01 CALL IDMSCOMP BEFORE STORE.
01-01 CALL IDMSCOMP BEFORE MODIFY.
01-01 CALL IDMSUCOM AFTER GET.
01-01 COPY BILLET-AA RECORD.
01-01
01-01 SET DESCRIPTION.
01-01
01-01
008000* *****
008100* * SET DESCRIPTION STATEMENTS *
008200* *****
008300*
008400 SET NAME IS IXSUB-IXREC.
008500 ORDER IS NEXT.
008600 MODE IS CHAIN LINKED TO PRIOR.
008700 OWNER IS IXSUB NEXT DBKEY POSITION IS 1.
008800 PRIOR DBKEY POSITION IS 2.
008900 MEMBER IS IXREC NEXT DBKEY POSITION IS 1.
009000 PRIOR DBKEY POSITION IS 2.
009100 MANDATORY AUTOMATIC.
009200
009300 SET NAME IS IXSET-IXREC.
009400 ORDER IS NEXT.
009500 MODE IS CHAIN LINKED TO PRIOR.
009600 OWNER IS IXSET NEXT DBKEY POSITION IS 1.
009700 PRIOR DBKEY POSITION IS 2.
009800 MEMBER IS IXREC NEXT DBKEY POSITION IS 3.
009900 PRIOR DBKEY POSITION IS 4.
010000 MANDATORY AUTOMATIC.
010100
010200 SET NAME IS IXSET-IXDET.
010300 ORDER IS NEXT.
010400 MODE IS CHAIN LINKED TO PRIOR.
010500 OWNER IS IXSET NEXT DBKEY POSITION IS 3.
010600 PRIOR DBKEY POSITION IS 4.
010700 MEMBER IS IXDET NEXT DBKEY POSITION IS 1.
010800 PRIOR DBKEY POSITION IS 2.
010900 MANDATORY AUTOMATIC.
011000
01-01
01-01
01-01 SET NAME IS IX-ORGAN-UIC.
01-01 ORDER IS SORTED.

```

01-01 MODE IS CHAIN.  
 01-01 OWNER IS IXOWNER  
 01-01 NEXT DBKEY POSITION IS 1.  
 01-01 MEMBER IS ORGANIZATION  
 01-01 NEXT DBKEY POSITION IS 6  
 01-01 OPTIONAL MANUAL  
 01-01 ASCENDING KEY IS ORGAN-UIC DUPLICATES LAST.  
 01-01 SET NAME IS IX-ORGAN-CMDCODE.  
 01-01 ORDER IS SORTED.  
 01-01 MODE IS CHAIN.  
 01-01 OWNER IS IXOWNER  
 01-01 NEXT DBKEY POSITION IS 2.  
 01-01 MEMBER IS ORGANIZATION  
 01-01 NEXT DBKEY POSITION IS 5  
 01-01 OPTIONAL MANUAL  
 01-01 ASCENDING KEY IS ORGAN-CMDCODE DUPLICATES LAST.  
 01-01 SET NAME IS IX-PEOPLE-NAME.  
 01-01 ORDER IS SORTED.  
 01-01 MODE IS CHAIN.  
 01-01 OWNER IS IXOWNER  
 01-01 NEXT DBKEY POSITION IS 3.  
 01-01 MEMBER IS PEOPLE  
 01-01 NEXT DBKEY POSITION IS 5  
 01-01 OPTIONAL MANUAL  
 01-01 ASCENDING KEY IS NAME-PERSON DUPLICATES LAST.  
 01-01 SET NAME IS IX-LOC-COUNTRY.  
 01-01 ORDER IS SORTED.  
 01-01 MODE IS CHAIN.  
 01-01 OWNER IS IXOWNER  
 01-01 NEXT DBKEY POSITION IS 4.  
 01-01 MEMBER IS UDB-LOCATION  
 01-01 NEXT DBKEY POSITION IS 5  
 01-01 OPTIONAL MANUAL  
 01-01 ASCENDING KEY IS LOCATION-CNTRY DUPLICATES LAST.  
 01-01 SET NAME IS UDB-1000-A.  
 01-01 ORDER SORTED.  
 01-01 MODE IS CHAIN LINKED TO PRIOR.  
 01-01 OWNER IS ORGANIZATION  
 01-01 NEXT DBKEY POSITION IS 1  
 01-01 PRIOR DBKEY POSITION IS 2.  
 01-01 MEMBER IS ORG-NIMIS  
 01-01 NEXT DBKEY POSITION IS 1  
 01-01 PRIOR DBKEY POSITION IS 2  
 01-01 OWNER DBKEY POSITION IS 3  
 01-01 LINKED TO OWNER  
 01-01 ASCENDING KEY IS ID-ORG-NIMIS DUPLICATES FIRST  
 01-01 MANDATORY AUTOMATIC.  
 01-01 MEMBER IS SSO-CONTROL  
 01-01 NEXT DBKEY POSITION IS 1  
 01-01 PRIOR DBKEY POSITION IS 2  
 01-01 OWNER DBKEY POSITION IS 3  
 01-01 LINKED TO OWNER  
 01-01 ASCENDING KEY IS ID-SSO-CONTROL DUPLICATES FIRST  
 01-01 MANDATORY AUTOMATIC.  
 01-01 MEMBER IS BADGE  
 01-01 NEXT DBKEY POSITION IS 1  
 01-01 PRIOR DBKEY POSITION IS 2  
 01-01 OWNER DBKEY POSITION IS 3

01-01 LINKED TO OWNER  
 01-01 ASCENDING KEY IS ID-BADGE DUPLICATES NOT  
 01-01 MANDATORY AUTOMATIC.  
 01-01  
 01-01 SET NAME IS UDB-1000-B.  
 01-01 ORDER IS NEXT.  
 01-01 MODE IS CHAIN LINKED TO PRIOR.  
 01-01 OWNER IS ORGANIZATION  
 01-01 NEXT DBKEY POSITION IS 3  
 01-01 PRIOR DBKEY POSITION IS 4.  
 01-01 MEMBER IS SSO-CONTROL  
 01-01 NEXT DBKEY POSITION IS 4  
 01-01 PRIOR DBKEY POSITION IS 5  
 01-01 OWNER DBKEY POSITION IS 6  
 01-01 LINKED TO OWNER  
 01-01 MANDATORY MANUAL.  
 01-01 MEMBER IS LOCATION-ORGAN  
 01-01 NEXT DBKEY POSITION IS 4  
 01-01 PRIOR DBKEY POSITION IS 5  
 01-01 OWNER DBKEY POSITION IS 6  
 01-01 LINKED TO OWNER  
 01-01 MANDATORY AUTOMATIC.  
 01-01 MEMBER IS ORGAN-BILLET  
 01-01 NEXT DBKEY POSITION IS 1  
 01-01 PRIOR DBKEY POSITION IS 2  
 01-01 OWNER DBKEY POSITION IS 3  
 01-01 LINKED TO OWNER  
 01-01 MANDATORY AUTOMATIC.  
 01-01 MEMBER IS ORGAN-VISIT  
 01-01 NEXT DBKEY POSITION IS 1  
 01-01 PRIOR DBKEY POSITION IS 2  
 01-01 OWNER DBKEY POSITION IS 3  
 01-01 LINKED TO OWNER  
 01-01 MANDATORY AUTOMATIC.  
 01-01 MEMBER IS ORGAN-PEOPLE  
 01-01 NEXT DBKEY POSITION IS 1  
 01-01 PRIOR DBKEY POSITION IS 2  
 01-01 OWNER DBKEY POSITION IS 3  
 01-01 LINKED TO OWNER  
 01-01 MANDATORY AUTOMATIC.  
 01-01  
 01-01 SET NAME IS UDB-1003-A.  
 01-01 ORDER IS SORTED.  
 01-01 MODE IS CHAIN LINKED TO PRIOR.  
 01-01 OWNER IS PEOPLE  
 01-01 NEXT DBKEY POSITION IS 1  
 01-01 PRIOR DBKEY POSITION IS 2.  
 01-01 MEMBER IS PEOPLE-NIMIS  
 01-01 NEXT DBKEY POSITION IS 1  
 01-01 PRIOR DBKEY POSITION IS 2  
 01-01 OWNER DBKEY POSITION IS 3  
 01-01 LINKED TO OWNER  
 01-01 ASCENDING KEY IS ID-PEOPLE-NIMIS DUPLICATES FIRST  
 01-01 MANDATORY AUTOMATIC.  
 01-01 MEMBER IS PEOPLE-SMS  
 01-01 NEXT DBKEY POSITION IS 1  
 01-01 PRIOR DBKEY POSITION IS 2  
 01-01 OWNER DBKEY POSITION IS 3  
 01-01 LINKED TO OWNER  
 01-01 ASCENDING KEY IS ID-PEOPLE-SMS DUPLICATES NOT  
 01-01 MANDATORY AUTOMATIC.



01-01 MEMBER IS PEOPLE-RELATE  
 01-01 NEXT DBKEY POSITION IS 1  
 01-01 PRIOR DBKEY POSITION IS 2  
 01-01 OWNER DBKEY POSITION IS 3  
 01-01 LINKED TO OWNER  
 01-01 ASCENDING KEY IS ID-PEOPLE-RELATE DUPLICATES FIRST  
 01-01 MANDATORY AUTOMATIC.  
 01-01  
 01-01 SET NAME IS UDB-1003-B.  
 01-01 ORDER NEXT.  
 01-01 MODE IS CHAIN LINKED TO PRIOR.  
 01-01 OWNER IS PEOPLE  
 01-01 NEXT DBKEY POSITION IS 3  
 01-01 PRIOR DBKEY POSITION IS 4.  
 01-01 MEMBER IS PEOPLE-RELATE  
 01-01 NEXT DBKEY POSITION IS 4  
 01-01 PRIOR DBKEY POSITION IS 5  
 01-01 OWNER DBKEY POSITION IS 6  
 01-01 LINKED TO OWNER  
 01-01 MANDATORY MANUAL.  
 01-01  
 01-01 SET NAME IS UDB-1202-A.  
 01-01 ORDER IS SORTED.  
 01-01 MODE IS CHAIN LINKED TO PRIOR.  
 01-01 OWNER IS UDB-LOCATION  
 01-01 NEXT DBKEY POSITION IS 1  
 01-01 PRIOR DBKEY POSITION IS 2.  
 01-01 MEMBER IS LOCATION-ACCESS  
 01-01 NEXT DBKEY POSITION IS 1  
 01-01 PRIOR DBKEY POSITION IS 2  
 01-01 OWNER DBKEY POSITION IS 3  
 01-01 LINKED TO OWNER  
 01-01 ASCENDING KEY IS ID-LOCATION-ACCESS DUPLICATES NOT  
 01-01 MANDATORY AUTOMATIC.  
 01-01 MEMBER IS LOCATION-RELATE  
 01-01 NEXT DBKEY POSITION IS 1  
 01-01 PRIOR DBKEY POSITION IS 2  
 01-01 OWNER DBKEY POSITION IS 3  
 01-01 LINKED TO OWNER  
 01-01 ASCENDING KEY IS ID-LOCATION-RELATE DUPLICATES FIRST  
 01-01 MANDATORY AUTOMATIC.  
 01-01 MEMBER IS VISITOR-LOG  
 01-01 NEXT DBKEY POSITION IS 3  
 01-01 PRIOR DBKEY POSITION IS 4  
 01-01 OWNER DBKEY POSITION IS 5  
 01-01 LINKED TO OWNER  
 01-01 ASCENDING KEY IS ID-VISITOR-LOG DUPLICATES NOT  
 01-01 MANDATORY AUTOMATIC.  
 01-01  
 01-01 SET NAME IS UDB-1202-B.  
 01-01 ORDER IS NEXT.  
 01-01 MODE IS CHAIN LINKED TO PRIOR.  
 01-01 OWNER IS UDB-LOCATION  
 01-01 NEXT DBKEY POSITION IS 3  
 01-01 PRIOR DBKEY POSITION IS 4.  
 01-01 MEMBER IS LOCATION-RELATE  
 01-01 NEXT DBKEY POSITION IS 4  
 01-01 PRIOR DBKEY POSITION IS 5  
 01-01 OWNER DBKEY POSITION IS 6  
 01-01 LINKED TO OWNER



01-01 MANDATORY MANUAL.  
 01-01 MEMBER IS LOCATION-ORGAN  
 01-01 NEXT DBKEY POSITION IS 1  
 01-01 PRIOR DBKEY POSITION IS 2  
 01-01 OWNER DBKEY POSITION IS 3  
 01-01 LINKED TO OWNER  
 01-01 MANDATORY AUTOMATIC.  
 01-01 MEMBER IS CITIZENSHIP  
 01-01 NEXT DBKEY POSITION IS 4  
 01-01 PRIOR DBKEY POSITION IS 5  
 01-01 OWNER DBKEY POSITION IS 6  
 01-01 LINKED TO OWNER  
 01-01 MANDATORY AUTOMATIC.  
 01-01  
 01-01  
 01-01  
 01-01  
 01-01 SET NAME IS NIMIS-1004-A.  
 01-01 ORDER SORTED.  
 01-01 MODE IS CHAIN LINKED TO PRIOR.  
 01-01 OWNER IS NIM-PROJECT  
 01-01 NEXT DBKEY POSITION IS 1  
 01-01 PRIOR DBKEY POSITION IS 2.  
 01-01 MEMBER IS NIMPROJ-RELATE  
 01-01 NEXT DBKEY POSITION IS 1  
 01-01 PRIOR DBKEY POSITION IS 2  
 01-01 OWNER DBKEY POSITION IS 3  
 01-01 LINKED TO OWNER  
 01-01 ASCENDING KEY IS ID-NIMPROJ-RELATE DUPLICATES FIRST  
 01-01 MANDATORY AUTOMATIC.  
 01-01 MEMBER IS NIM-COMMENTS  
 01-01 NEXT DBKEY POSITION IS 1  
 01-01 PRIOR DBKEY POSITION IS 2  
 01-01 ASCENDING KEY IS ID-NIM-COMMENTS DUPLICATES NOT  
 01-01 MANDATORY AUTOMATIC.  
 01-01 MEMBER IS NIM-MSTONES  
 01-01 NEXT DBKEY POSITION IS 5  
 01-01 PRIOR DBKEY POSITION IS 6  
 01-01 OWNER DBKEY POSITION IS 7  
 01-01 LINKED TO OWNER  
 01-01 ASCENDING KEY IS ID-NIM-MSTONES DUPLICATES NOT  
 01-01 MANDATORY AUTOMATIC.  
 01-01 MEMBER IS NIM-SUBTASK  
 01-01 NEXT DBKEY POSITION IS 5  
 01-01 PRIOR DBKEY POSITION IS 6  
 01-01 OWNER DBKEY POSITION IS 7  
 01-01 LINKED TO OWNER  
 01-01 ASCENDING KEY IS ID-NIM-SUBTASK DUPLICATES NOT  
 01-01 MANDATORY AUTOMATIC.  
 01-01 MEMBER IS NIM-LABORERS  
 01-01 NEXT DBKEY POSITION IS 1  
 01-01 PRIOR DBKEY POSITION IS 2  
 01-01 OWNER DBKEY POSITION IS 3  
 01-01 LINKED TO OWNER  
 01-01 ASCENDING KEY IS ID-NIM-LABORERS DUPLICATES LAST  
 01-01 MANDATORY AUTOMATIC.  
 01-01  
 01-01 SET NAME IS NIMIS-1004-B.  
 01-01 ORDER NEXT.  
 01-01 MODE IS CHAIN LINKED TO PRIOR.

```

01-01 OWNER IS NIM-PROJECT
01-01 NEXT DBKEY POSITION IS 3.
01-01 PRIOR DBKEY POSITION IS 4.
01-01 MEMBER IS NIMPROJ-RELATE
01-01 NEXT DBKEY POSITION IS 4
01-01 PRIOR DBKEY POSITION IS 5
01-01 OWNER DBKEY POSITION IS 6
01-01 LINKED TO OWNER
01-01 MANDATORY MANUAL.
01-01 MEMBER IS ORG-NIMPROJ
01-01 NEXT DBKEY POSITION IS 4
01-01 PRIOR DBKEY POSITION IS 5
01-01 OWNER DBKEY POSITION IS 6
01-01 LINKED TO OWNER
01-01 MANDATORY AUTOMATIC.
01-01 SET NAME IS NIMIS-1006-A.
01-01 ORDER SORTED.
01-01 MODE IS CHAIN LINKED TO PRIOR.
01-01 OWNER IS POSIT-DESCR
01-01 NEXT DBKEY POSITION IS 9
01-01 PRIOR DBKEY POSITION IS 10.
01-01 MEMBER IS DUM-POSIT-DESCR
01-01 NEXT DBKEY POSITION IS 3
01-01 PRIOR DBKEY POSITION IS 4
01-01 ASCENDING KEY IS ID-DUM-POSIT-DESCR DUPLICATES NOT
01-01 OPTIONAL MANUAL.
01-01 SET NAME IS NIMIS-1006-B.
01-01 ORDER NEXT.
01-01 MODE IS CHAIN LINKED TO PRIOR.
01-01 OWNER IS POSIT-DESCR
01-01 NEXT DBKEY POSITION IS 7
01-01 PRIOR DBKEY POSITION IS 8.
01-01 MEMBER IS DUM-POSIT-DESCR
01-01 NEXT DBKEY POSITION IS 1
01-01 PRIOR DBKEY POSITION IS 2
01-01 OPTIONAL MANUAL.
01-01 SET NAME IS NIMIS-1009-A.
01-01 ORDER SORTED.
01-01 MODE IS CHAIN LINKED TO PRIOR.
01-01 OWNER IS NIM-MSTONES
01-01 NEXT DBKEY POSITION IS 1
01-01 PRIOR DBKEY POSITION IS 2.
01-01 MEMBER IS DUM-NIM-MSTONES
01-01 NEXT DBKEY POSITION IS 1
01-01 PRIOR DBKEY POSITION IS 2
01-01 ASCENDING KEY IS ID-DUM-NIM-MSTONES DUPLICATES NOT
01-01 OPTIONAL MANUAL.
01-01 MEMBER IS NIM-SUBTASK
01-01 NEXT DBKEY POSITION IS 8
01-01 PRIOR DBKEY POSITION IS 9
01-01 OWNER DBKEY POSITION IS 10
01-01 LINKED TO OWNER
01-01 ASCENDING KEY IS ID-NIM-SUBTASK DUPLICATES NOT
01-01 OPTIONAL AUTOMATIC.
01-01 MEMBER IS NIM-MSTONE-CMTS
01-01 NEXT DBKEY POSITION IS 1
01-01 PRIOR DBKEY POSITION IS 2
01-01 ASCENDING KEY IS ID-NIM-MSTONE-CMTS DUPLICATES NOT
01-01 MANDATORY AUTOMATIC.
01-01 SET NAME IS NIMIS-1009-B.

```

01-01 ORDER NEXT.  
 01-01 MODE IS CHAIN LINKED TO PRIOR.  
 01-01 OWNER IS NIM-MSTONES  
 01-01 NEXT DBKEY POSITION IS 3  
 01-01 PRIOR DBKEY POSITION IS 4.  
 01-01 MEMBER IS DUM-NIM-MSTONES  
 01-01 NEXT DBKEY POSITION IS 3  
 01-01 PRIOR DBKEY POSITION IS 4  
 01-01 OPTIONAL MANUAL.  
 01-01 MEMBER IS NIM-SUBTASK  
 01-01 NEXT DBKEY POSITION IS 11  
 01-01 PRIOR DBKEY POSITION IS 12  
 01-01 OWNER DBKEY POSITION IS 13  
 01-01 LINKED TO OWNER  
 01-01 OPTIONAL MANUAL.  
 01-01 SET NAME IS NIMIS-1010-B.  
 01-01 ORDER NEXT.  
 01-01 MODE IS CHAIN LINKED TO PRIOR.  
 01-01 OWNER IS DUM-NIM-MSTONES  
 01-01 NEXT DBKEY POSITION IS 5  
 01-01 PRIOR DBKEY POSITION IS 6.  
 01-01 MEMBER IS NIM-MSTONES  
 01-01 NEXT DBKEY POSITION IS 8  
 01-01 PRIOR DBKEY POSITION IS 9  
 01-01 OWNER DBKEY POSITION IS 10  
 01-01 LINKED TO OWNER  
 01-01 OPTIONAL MANUAL.  
 01-01 SET NAME IS NIMIS-1011-A.  
 01-01 ORDER SORTED.  
 01-01 MODE IS CHAIN LINKED TO PRIOR.  
 01-01 OWNER IS NIM-SUBTASK  
 01-01 NEXT DBKEY POSITION IS 1  
 01-01 PRIOR DBKEY POSITION IS 2.  
 01-01 MEMBER IS DUM-NIM-SUBTASK  
 01-01 NEXT DBKEY POSITION IS 1  
 01-01 PRIOR DBKEY POSITION IS 2  
 01-01 ASCENDING KEY IS ID-DUM-NIM-SUBTASK DUPLICATES NOT  
 01-01 OPTIONAL MANUAL.  
 01-01 MEMBER IS NIM-SUBTASK-CMTS  
 01-01 NEXT DBKEY POSITION IS 1  
 01-01 PRIOR DBKEY POSITION IS 2  
 01-01 ASCENDING KEY IS ID-NIM-SUBTASK-CMTS DUPLICATES NOT  
 01-01 MANDATORY AUTOMATIC.  
 01-01 SET NAME IS NIMIS-1011-B.  
 01-01 ORDER NEXT.  
 01-01 MODE IS CHAIN LINKED TO PRIOR.  
 01-01 OWNER IS NIM-SUBTASK  
 01-01 NEXT DBKEY POSITION IS 3  
 01-01 PRIOR DBKEY POSITION IS 4.  
 01-01 MEMBER IS DUM-NIM-SUBTASK  
 01-01 NEXT DBKEY POSITION IS 3  
 01-01 PRIOR DBKEY POSITION IS 4  
 01-01 OPTIONAL MANUAL.  
 01-01 SET NAME IS NIMIS-1012-A.  
 01-01 ORDER SORTED.  
 01-01 MODE IS CHAIN LINKED TO PRIOR.  
 01-01 OWNER IS DUM-NIM-SUBTASK  
 01-01 NEXT DBKEY POSITION IS 5  
 01-01 PRIOR DBKEY POSITION IS 6.

01-01 MEMBER IS NIM-SUBTASK  
 01-01 NEXT DBKEY POSITION IS 14  
 01-01 PRIOR DBKEY POSITION IS 15  
 01-01 OWNER DBKEY POSITION IS 15  
 01-01 LINKED TO OWNER  
 01-01 ASCENDING KEY IS ID-NIM-SUBTASK DUPLICATES NOT  
 01-01 OPTIONAL MANUAL.  
 01-01 SET NAME IS NIMIS-1013-B.  
 01-01 ORDER IS NEXT.  
 01-01 MODE IS CHAIN LINKED TO PRIOR.  
 01-01 OWNER IS PEOPLE-NIMIS  
 01-01 NEXT DBKEY POSITION IS 4  
 01-01 PRIOR DBKEY POSITION IS 5.  
 01-01 MEMBER IS POSIT-DESCR  
 01-01 NEXT DBKEY POSITION IS 4  
 01-01 PRIOR DBKEY POSITION IS 5  
 01-01 OWNER DBKEY POSITION IS 6  
 01-01 LINKED TO OWNER  
 01-01 OPTIONAL MANUAL.  
 01-01 MEMBER IS NIM-LABORERS  
 01-01 NEXT DBKEY POSITION IS 4  
 01-01 PRIOR DBKEY POSITION IS 5  
 01-01 OWNER DBKEY POSITION IS 6  
 01-01 LINKED TO OWNER  
 01-01 MANDATORY AUTOMATIC.  
 01-01  
 01-01 SET NAME IS NIMIS-1019-A.  
 01-01 ORDER SORTED.  
 01-01 MODE IS CHAIN LINKED TO PRIOR.  
 01-01 OWNER IS ORG-NIMIS  
 01-01 NEXT DBKEY POSITION IS 4  
 01-01 PRIOR DBKEY POSITION IS 5.  
 01-01 MEMBER IS POSIT-DESCR  
 01-01 NEXT DBKEY POSITION IS 1  
 01-01 PRIOR DBKEY POSITION IS 2  
 01-01 OWNER DBKEY POSITION IS 3  
 01-01 LINKED TO OWNER  
 01-01 ASCENDING KEY IS ID-POSIT-DESCR DUPLICATES NOT  
 01-01 MANDATORY AUTOMATIC.  
 01-01 MEMBER IS ORG-COMMENTS  
 01-01 NEXT DBKEY POSITION IS 1  
 01-01 PRIOR DBKEY POSITION IS 2  
 01-01 ASCENDING KEY IS ID-ORGAN-COMMENTS DUPLICATES NOT  
 01-01 MANDATORY AUTOMATIC.  
 01-01  
 01-01 SET NAME IS NIMIS-1019-B.  
 01-01 ORDER NEXT.  
 01-01 MODE IS CHAIN LINKED TO PRIOR.  
 01-01 OWNER IS ORG-NIMIS  
 01-01 NEXT DBKEY POSITION IS 6  
 01-01 PRIOR DBKEY POSITION IS 7.  
 01-01 MEMBER IS ORG-NIMPROJ  
 01-01 NEXT DBKEY POSITION IS 1  
 01-01 PRIOR DBKEY POSITION IS 2  
 01-01 OWNER DBKEY POSITION IS 3  
 01-01 LINKED TO OWNER  
 01-01 MANDATORY AUTOMATIC.  
 01-01  
 01-01 SET NAME IS SMS-1200-A.  
 01-01 ORDER IS SORTED.



01-01 MODE IS CHAIN LINKED TO PRIOR.  
 01-01 OWNER IS BILLET  
 01-01 NEXT DBKEY POSITION IS 3  
 01-01 PRIOR DBKEY POSITION IS 4.  
 01-01 MEMBER IS ACCESS-BILLET  
 01-01 NEXT DBKEY POSITION IS 4  
 01-01 PRIOR DBKEY POSITION IS 5  
 01-01 OWNER DBKEY POSITION IS 6  
 01-01 LINKED TO OWNER  
 01-01 ASCENDING KEY IS ID-ACCESS-BILLET DUPLICATES NOT  
 01-01 MANDATORY AUTOMATIC.  
 01-01  
 01-01 SET NAME IS SMS-1200-B.  
 01-01 ORDER IS NEXT.  
 01-01 MODE IS CHAIN LINKED TO PRIOR.  
 01-01 OWNER IS BILLET  
 01-01 NEXT DBKEY POSITION IS 1  
 01-01 PRIOR DBKEY POSITION IS 2.  
 01-01 MEMBER IS ORGAN-BILLET  
 01-01 NEXT DBKEY POSITION IS 4  
 01-01 PRIOR DBKEY POSITION IS 5  
 01-01 OWNER DBKEY POSITION IS 6  
 01-01 LINKED TO OWNER  
 01-01 MANDATORY AUTOMATIC.  
 01-01 MEMBER IS BILLET-PEOPLE  
 01-01 NEXT DBKEY POSITION IS 1  
 01-01 PRIOR DBKEY POSITION IS 2  
 01-01 OWNER DBKEY POSITION IS 3  
 01-01 LINKED TO OWNER  
 01-01 MANDATORY AUTOMATIC.  
 01-01  
 01-01 MEMBER IS BILLET-AA  
 01-01 NEXT DBKEY POSITION IS 1  
 01-01 PRIOR DBKEY POSITION IS 2  
 01-01 OWNER DBKEY POSITION IS 3  
 01-01 LINKED TO OWNER  
 01-01 MANDATORY AUTOMATIC.  
 01-01  
 01-01 SET NAME IS SMS-1201-A.  
 01-01 ORDER IS SORTED.  
 01-01 MODE IS CHAIN LINKED TO PRIOR.  
 01-01 OWNER IS ACCESS-TYPE  
 01-01 NEXT DBKEY POSITION IS 1  
 01-01 PRIOR DBKEY POSITION IS 2.  
 01-01 MEMBER IS ACCESS-TYPE-TYPE  
 01-01 NEXT DBKEY POSITION IS 1  
 01-01 PRIOR DBKEY POSITION IS 2  
 01-01 OWNER DBKEY POSITION IS 3  
 01-01 LINKED TO OWNER  
 01-01 ASCENDING KEY IS ID-ACCESS-TYPE-TYPE DUPLICATES FIRST  
 01-01 MANDATORY AUTOMATIC.  
 01-01  
 01-01 SET NAME IS SMS-1201-B.  
 01-01 ORDER IS NEXT.  
 01-01 MODE IS CHAIN LINKED TO PRIOR.  
 01-01 OWNER IS ACCESS-TYPE  
 01-01 NEXT DBKEY POSITION IS 3  
 01-01 PRIOR DBKEY POSITION IS 4.  
 01-01 MEMBER IS LOCATION-ACCESS



```

01-01      NEXT DBKEY POSITION IS 4
01-01      PRIOR DBKEY POSITION IS 5
01-01      OWNER DBKEY POSITION IS 6
01-01      LINKED TO OWNER
01-01      MANDATORY AUTOMATIC.
01-01      MEMBER IS ACCESS-TYPE-TYPE
01-01      NEXT DBKEY POSITION IS 4
01-01      PRIOR DBKEY POSITION IS 5
01-01      OWNER DBKEY POSITION IS 6
01-01      LINKED TO OWNER
01-01      MANDATORY MANUAL.
01-01      MEMBER IS ACCESS-BULLET
01-01      NEXT DBKEY POSITION IS 1
01-01      PRIOR DBKEY POSITION IS 2
01-01      OWNER DBKEY POSITION IS 3
01-01      LINKED TO OWNER
01-01      MANDATORY AUTOMATIC.
01-01      MEMBER IS ACCESS-ASSIGNED
01-01      NEXT DBKEY POSITION IS 1
01-01      PRIOR DBKEY POSITION IS 2
01-01      OWNER DBKEY POSITION IS 3
01-01      LINKED TO OWNER
01-01      MANDATORY AUTOMATIC.
01-01
01-01      SET NAME IS SMS-1203-A.
01-01      ORDER IS SORTED.
01-01      MODE IS CHAIN LINKED TO PRIOR.
01-01      OWNER IS LOCATION-ACCESS
01-01      NEXT DBKEY POSITION IS 7
01-01      PRIOR DBKEY POSITION IS 8.
01-01      MEMBER IS LOC-ACCESS-CMT
01-01      NEXT DBKEY POSITION IS 1
01-01      PRIOR DBKEY POSITION IS 2
01-01      OWNER DBKEY POSITION IS 3
01-01      LINKED TO OWNER
01-01      ASCENDING KEY IS ID-LOC-ACCESS-CMT DUPLICATES NOT
01-01      MANDATORY AUTOMATIC.
01-01
01-01      SET NAME IS SMS-1211-B.
01-01      ORDER IS NEXT.
01-01      MODE IS CHAIN LINKED TO PRIOR.
01-01      OWNER IS VISITOR-LOG
01-01      NEXT DBKEY POSITION IS 1
01-01      PRIOR DBKEY POSITION IS 2.
01-01      MEMBER IS ORGAN-VISIT
01-01      NEXT DBKEY POSITION IS 4
01-01      PRIOR DBKEY POSITION IS 5
01-01      OWNER DBKEY POSITION IS 6
01-01      LINKED TO OWNER
01-01      MANDATORY AUTOMATIC.
01-01      MEMBER IS PEOPLE-VISIT
01-01      NEXT DBKEY POSITION IS 4
01-01      PRIOR DBKEY POSITION IS 5
01-01      OWNER DBKEY POSITION IS 6
01-01      LINKED TO OWNER
01-01      MANDATORY AUTOMATIC.
01-01
01-01      SET NAME IS SMS-1216-F.
01-01      ORDER NEXT.
01-01      MODE IS CHAIN LINKED TO PRIOR.

```

01-01 OWNER IS ACCESS-ASSIGNED  
 01-01 NEXT DBKEY POSITION IS 7  
 01-01 PRIOR DBKEY POSITION IS 8.  
 01-01 MEMBER IS BILLET-AA  
 01-01 NEXT DBKEY POSITION IS 4  
 01-01 PRIOR DBKEY POSITION IS 5  
 01-01 OWNER DBKEY POSITION IS 6  
 01-01 LINKED TO OWNER  
 01-01 MANDATORY AUTOMATIC.  
 01-01  
 01-01 SET NAME IS SMS-1219-A.  
 01-01 ORDER IS SORTED.  
 01-01 MODE IS CHAIN LINKED TO PRIOR.  
 01-01 OWNER IS PEOPLE-SMS  
 01-01 NEXT DBKEY POSITION IS 4  
 01-01 PRIOR DBKEY POSITION IS 5.  
 01-01 MEMBER IS ACCESS-ASSIGNED  
 01-01 NEXT DBKEY POSITION IS 4  
 01-01 PRIOR DBKEY POSITION IS 5  
 01-01 OWNER DBKEY POSITION IS 6  
 01-01 LINKED TO OWNER  
 01-01 ASCENDING KEY IS ID-ACCESS-ASSIGNED DUPLICATES NOT  
 01-01 MANDATORY AUTOMATIC.  
 01-01  
 01-01 SET NAME IS SMS-1219-B.  
 01-01 ORDER NEXT.  
 01-01 MODE IS CHAIN LINKED TO PRIOR.  
 01-01 OWNER IS PEOPLE-SMS  
 01-01 NEXT DBKEY POSITION IS 6  
 01-01 PRIOR DBKEY POSITION IS 7.  
 01-01 MEMBER IS PEOPLE-VISIT  
 01-01 NEXT DBKEY POSITION IS 1  
 01-01 PRIOR DBKEY POSITION IS 2  
 01-01 OWNER DBKEY POSITION IS 3  
 01-01 LINKED TO OWNER  
 01-01 MANDATORY AUTOMATIC.  
 01-01 MEMBER IS ORGAN-PEOPLE  
 01-01 NEXT DBKEY POSITION IS 4  
 01-01 PRIOR DBKEY POSITION IS 5  
 01-01 OWNER DBKEY POSITION IS 6  
 01-01 LINKED TO OWNER  
 01-01 MANDATORY AUTOMATIC.  
 01-01 MEMBER IS CITIZENSHIP  
 01-01 NEXT DBKEY POSITION IS 1  
 01-01 PRIOR DBKEY POSITION IS 2  
 01-01 OWNER DBKEY POSITION IS 3  
 01-01 LINKED TO OWNER  
 01-01 MANDATORY AUTOMATIC.  
 01-01 MEMBER IS BILLET-PEOPLE  
 01-01 NEXT DBKEY POSITION IS 4  
 01-01 PRIOR DBKEY POSITION IS 5  
 01-01 OWNER DBKEY POSITION IS 6  
 01-01 LINKED TO OWNER  
 01-01 MANDATORY AUTOMATIC.  
 01-01 MEMBER IS BADGE  
 01-01 NEXT DBKEY POSITION IS 4  
 01-01 PRIOR DBKEY POSITION IS 5  
 01-01 OWNER DBKEY POSITION IS 6  
 01-01 LINKED TO OWNER  
 01-01 OPTIONAL MANUAL.  
 01-01

## APPENDIX I

### GLOSSARY OF TERMS

This glossary of terms is limited to those terms which are used in the course of this Guide. Words which are defined elsewhere within the glossary are displayed in capital letters.

#### A

AREA - The logical definition of the space within the database where record entities will occur. Areas are divided into a specified number of PAGES.

ATTRIBUTE - Within the IDD, an attribute is a specific word or term which further describes the dictionary entry. Each attribute is associated with a CLASS, e.g., "ENTRY-SECURITY IS ENTRY-UNCLASSIFIED" where "ENTRY-SECURITY" is a CLASS and "ENTRY-UNCLASSIFIED" is an attribute within the class.

AUTOMATIC - The term, as applied to IDMS, means a type of SET where a member record is automatically associated with its owner at the time the member record is stored in the database.

#### B

BASE RECORD - A UDB record which is stored CALC in the database and which has no owner records. A primary database entry point.

BATCH - The term used to describe ADP operation which is characterized by the use of punched card input and printed report output.

#### C

CALC - The term which identifies an IDMS record type as being stored randomly in the database in a location determined by the contents of its KEY elements.

CAMP - The Central Access Monitor Program of IDMS. This program is essential for central version (CV) operation.

CLASS - A basic grouping of words or terms which provide further explanation and organizing of entries in the data dictionary. See also: ATTRIBUTE.

CULPRIT - A report writer and batch query facility marketed by Cullinane Corporation which interfaces with IDMS.

CV - Central Version. The control program which permits multiple users of IDMS to share DBMS resources and prevent concurrent updating of the database by multiple users.

## D

DATA BASE - A description of the function of utilizing DBMS for the maintenance of information.

DATABASE - The physical repository of information processed by a DBMS.

DATA ELEMENT - The lowest level of data definition within the database. A data element contains a specific piece of information which cannot be logically subdivided.

DATA ENTITY - The highest level of data definition within the database. A data entity is the logical association of a number of DATA ELEMENTS and DATA GROUPS to form a complete picture of some information. See also: RECORD.

DATA GROUP - An intermediate level of data definition within the database. A data group consists of one or more DATA ELEMENTS and/or other data groups which together associate logically-related information.

DATA SET - The physical storage space set aside by the operating system for storing a database.

DBMS - Data Base Management System.

## E

ENTITY - See DATA ENTITY

## F

FILE - See AREA.

FIXED-LENGTH ELEMENT. A data element which will always contain sufficient data to completely fill its storage space, e.g., country code is a 2-byte field and every entry to the element is always 2 bytes long.

## G

GROUP - See DATA GROUP.

## H

## I

IDD - Integrated Data Dictionary. The software package, marketed by Cullinane Corporation, which maintains the IDMS data dictionary.

IDMS - Integrated Database Management System.

IP - Input Processing. The acronym is normally used in conjunction with software developed to perform input processing functions supporting the integrated database.

INTERMEDIATE RECORD - One of the four UDB record format types. The intermediate record is characterized by its use as a relational intersection between two other record types or, in some cases only one, and its use as an owner of other record types.

## J

JCL - Job Control Language. The punched card deck which controls execution of IBM 360 software.

JOURNAL - The IDMS file which maintains database recovery information.



## K

KEY - A data element or group of data elements which defines a database record uniquely.

## L

LOGICAL RECORD - A record definition which is not concerned with the conditions of its physical use. The logical record, for example, does not consider the AREA in which it resides or the POINTERS which link it to other records. See also: DATA ENTITY.

## M

MANDATORY - The record occurrence must be associated with an owner record occurrence. Once stored, the association cannot be broken.

MANUAL - The association of two record occurrences whose record types are joined by a set is determined by the logic of supporting software.

MEMBER - A logical database record type which is dependently associated with another record type, its OWNER.

## N

## O

OLQ - The On-Line Query software package marketed by Cullinane Corporation which permits querying the IDMS database from remote interactive terminals.

OPTIONAL - The record occurrence, once stored in the database, may be transferred from the association with one owner record occurrence to another.

OWNER - A logical record type which has dependent MEMBER records associated with it.

## P

PAGE - The logical subdivision of a database AREA. Pages are assigned physical sizes when they are defined within a DATA SET.

## Q

## R

RECORD - The logical or physical representation of data within the database. The SCHEMA is composed of many RECORD TYPES, each of which may have an unlimited number of occurrences. These occurrences are generally referred to as records.

RECORD FORMAT - One of the general formats use to aid in the design of a RECORD TYPE. Four record formats are used by UDB.

RECORD TYPE - The individual definition of a record within the database SCHEMA.

RELATIONAL RECORD - A UDB record type which is used to associate two other record types.

RJE - Remote Job Entry terminal, used primarily for batch processing.

## S

SCHEMA - The logical definition of an entire database. The schema is the road map to the database.

SERVICE ANALYSIS - The process of identifying each function which will support a user's application, including database maintenance, reporting, and querying.

SET - The term used in IDMS to identify the association between two or more record types.

SUBSCHEMA - A subset of a SCHEMA which identifies a specific window view of the database, restricting the user to that area of the database.

SUBSIDIARY RECORD - A UDB record type which is the lowest level record to be defined. A subsidiary record may not be an OWNER record.

#### T

TERMINAL - Within this Guide, a terminal is meant to define a device which can communicate with the integrated database, primarily one which can communicate interactively.

TP - Teleprocessing.

#### U

UDB - The acronym for the Universal DataBase. The UDB is a concept which defines database structures in such a way that they can be readily updated without requiring extensive time and labor resources.

USER - Within this Guide, the term defines the persons or organizations which will utilize the integrated database.

#### V

VARIABLE-LENGTH ELEMENT - A data element which is large enough to contain the information which it is designed to hold but which may have empty space (spaces or zeros) after the data has been loaded; e.g., the name of a person is 27 bytes long but the data loaded may be only 15 bytes long.

## APPENDIX J.1

### PSL UPDATE

This appendix defines the syntax of OS/MVT Job Control Language (JCL) required to update the PSL members supporting database development. Appendix A.2 describes preparation of the contents of PSL members.

```
//ABBBBBBCC JOB (DEEE,FFFF,GG,HH,,,,,II),'PSL UPDATE',  
//  MSGLEVEL=(0,0),CLASS=E,REGION=60K  
//  EXEC PGM=IEBUPDTE  
//SYSPRINT DD SYSOUT=A  
//SYSUT1 DD DSN=DEV.PSL.XXX,DISP=OLD  
//SYSUT2 DD DSN=DEV.PSL.XXX,DISP=OLD  
//SYSIN DD *
```

PLACE UPDATE CARDS HERE

```
./ ENDUP  
/*  
//
```

Figure J1.1

Parameter definitions:

A	Job prefix, assigned by project manager
BBBBB	Development work order number, assigned by project manager
CC	Unique job identifier, assigned by user
D	Building designator, assigned by project manager
EEE	Person identifier, assigned by project manager
FFFF	Room number or other routing indicator, assigned by user
GG	Estimated execution time in minutes
HH	Estimate print lines in thousands
II	Number of print lines on a page, normally 60
XXX	Individual PSL name, assigned by project manager

## APPENDIX J.2

### IP COMPILATION

This appendix defines the syntax of OS/MVT Job Control Language (JCL) required to compile input processing (IP) programs for batch and on-line data entry support.

#### COMPILE BATCH IP FROM PARAMETER CARDS

```
//ABBBBCC JOB (DEEE,FFFF,GG,HH,,,,,II),'IP FROM CARDS',  
//  MSGLEVEL=(0,0),CLASS=E,REGION=150K  
//  EXEC ZIPXLKED,PHASE=IPZZZS,DUM2=  
//EXT.SYSIN      DD  *
```

#### PLACE IP PARAMETER CARDS HERE

```
//DML5.SYSIPT DD DSN=&&PSLOUT,DISP=(OLD,DELETE)  
/*  
//
```



Figure J2.1

Parameter definitions:

A	Job prefix, assigned by project manager
BBBBB	Development work order number, assigned by project manager
CC	Unique job identifier, assigned by user
D	Building designator, assigned by project manager
EEE	Person identifier, assigned by project manager
FFFF	Room number or other routing indicator, assigned by user
GG	Estimated execution time in minutes
HH	Estimate print lines in thousands
II	Number of print lines on a page, normally 60
ZZZ	IP primary identifier, assigned by project manager
S	IP option, S-store, M-modify, D-delete

COMPILE BATCH IP FROM PSL

```
//ABBBBBBCC JOB (DEEE,FFFF,GG,HH,,,,,II),'IP FROM PSL',  
// MSGLEVEL=(0,0),CLASS=E,REGION=150K  
// EXEC ZIPXLKED,PHASE=IPZZZS,DUM2=  
//EXT.SYSIN DD DSN=DEV.PSL.XXX(GCYZZZS),DISP=SHR  
//DML5.SYSIPT DD DSN=&&PSLOUT,DISP=(OLD,DELETE)  
/*  
//
```

Figure J2.2

Parameter definitions:

A	Job prefix, assigned by project manager
BBBBB	Development work order number, assigned by project manager
CC	Unique job identifier, assigned by user
D	Building designator, assigned by project manager
EEE	Person identifier, assigned by project manager
FFFF	Room number or other routing indicator, assigned by user
GG	Estimated execution time in minutes
HH	Estimate print lines in thousands
II	Number of print lines on a page, normally 60
XXX	Individual development PSL file, assigned by project manager
ZZZ	IP primary identifier, assigned by project manager
S	IP option, S-store, M-modify, D-delete

COMPILE ONLINE IP FROM PARAMETER CARDS

```
//ABBBBBBCC JOB (DEEE,FFFF,GG,HH,,,,,II),'CREATE ONLINE IP',  
//  MSGLEVEL=(0,0),CLASS=E,REGION=200K  
//  EXEC OLGEN,PHASE=ZZZS  
//GEN.SYSIN      DD  *
```

PLACE IP PARAMETER CARDS HERE

```
/*  
//
```

Figure J2.3

Parameter definitions:

A	Job prefix, assigned by project manager
BBBBB	Development work order number, assigned by project manager
CC	Unique job identifier, assigned by user
D	Building designator, assigned by project manager
EEE	Person identifier, assigned by project manager
FFFF	Room number or other routing indicator, assigned by user
GG	Estimated execution time in minutes
HH	Estimate print lines in thousands
II	Number of print lines on a page, normally 60
ZZZ	IP primary identifier, assigned by project manager
S	IP option, S-store, M-modify, D-delete

### APPENDIX J.3

#### CULPRIT REPORTS

This appendix defines the syntax of OS/MVT Job Control Language (JCL) required to prepare CULPRIT reports supporting batch processing and data conversion.

```
//ABBBBCC JOB (DEEE,FFFF,GG,HH,,,,,II),'CULPRIT REPORTS',  
//  MSGLEVEL=(0,0),CLASS=E,REGION=150K  
//  EXEC CULPRIT5  
//CULP0.SYSIN      DD  *
```

PLACE CULPRIT PARAMETER CARDS HERE

```
./ ENDUP  
/*  
//
```

Figure J3.1

Parameter definitions:

A	Job prefix, assigned by project manager
BBBBB	Development work order number, assigned by project manager
CC	Unique job identifier, assigned by user
D	Building designator, assigned by project manager
EEE	Person identifier, assigned by project manager
FFFF	Room number or other routing indicator, assigned by user
GG	Estimated execution time in minutes
HH	Estimate print lines in thousands
II	Number of print lines on a page, normally 60

## APPENDIX J.4

### IP EXECUTION

This appendix defines the syntax of OS/MVT Job Control Language (JCL) required to update the database in a batch environment.

```
//ABBBBBBCC JOB (DEEE,FFFF,GG,HH,,,,,II),'IP EXECUTION',  
//  MSGLEVEL=(0,0),CLASS=E,REGION=150K  
//  EXEC LDRUPDT5,PROG=LDR5  
//LDR.TRANSINP      DD  *
```

PLACE CTLA CONTROL CARD HERE  
PLACE DATA CARDS HERE

```
./ ENDUP  
/*  
//
```

Figure J4.1

Parameter definitions:

A	Job prefix, assigned by project manager
BBBBB	Development work order number, assigned by project manager
CC	Unique job identifier, assigned by user
D	Building designator, assigned by project manager
EEE	Person identifier, assigned by project manager
FFFF	Room number or other routing indicator, assigned by user
GG	Estimated execution time in minutes
HH	Estimate print lines in thousands
II	Number of print lines on a page, normally 60



## APPENDIX K

### INTEGRATED DATABASE DEVELOPMENT SCHEDULE AND DELIVERABLES

The NIPSSA Integrated Database Development and Design Guide, Version 2 defines the procedures to be followed during the development of a database application. This document identifies the steps and deliverable products which are to be produced during each phase of database development.

It is assumed that contractor personnel will be the primary external assistance supporting database development. References to programmers and analysts within this document are to contractor personnel unless specifically stated otherwise.

#### GENERAL DELIVERABLE PROCEDURE

The POAM is divided into 27 specific steps, each with a function and end product. Four of the steps are performed specifically by NIPSSA (3, 5, 7, and 8). The remainder are performed by the contractor.

The overall development effort is divided into 7 phases, five performed entirely by the contractor (1, 4, 5, 6, and 7). Phase 2 is performed jointly by the contractor and NIPSSA in series. Phase 3 is performed entirely by NIPSSA.

To simplify accounting and achieve flexibility within the development, formal contractor product deliveries will occur at the end of each phase except Phase 3 (performed by NIPSSA). The deliverables defined at the end of each step within a phase will be formally presented to NIPSSA at the end of the Phase. Delivery dates within each phase will be viewed as milestone reviews to maintain visibility of project progress. THE CONTRACTOR MUST UNDERSTAND THAT STEP DELIVERABLES FOLLOWED BY NIPSSA REVIEWS MUST REACH NIPSSA AT THE DEFINED DATE IN ORDER FOR NIPSSA TO RESPOND WITHIN THE DEFINED TIME FRAME.

#### ACCEPTANCE OF DELIVERABLES

Formal deliverable products due at the end of Phases 1, 2, 4, 5, 6, and 7 must be received by NIPSSA on the defined date. Receipt by intermediary organizations on the

delivery date does not constitute on-time delivery.

Formal deliverable products must be delivered in the form and format specified in the NIPSSA Database Development and Design Guide, Version 2 or this POAM. Unless previously agreed in writing, other forms of delivery are not acceptable.

Formal acceptance or rejection of deliverable products will be provided by NIPSSA in writing to the contractor or intermediate organization, as appropriate.

#### 1. PHASE I. PROBLEM/APPLICATION DEFINITION.

Phase I begins with the receipt of a user request for database support. This request should be received as defined by NAVINTCOM Instruction 5230.4C of 26 September 1978.

The development of a new database facility requires that a subsystem specification be prepared. Appendix D of the Design Guide provides guidelines for the content of the subsystem specification which is stored on a Program Support Library specified by the NIPSSA project manager.

The contractor will interview end user personnel to determine:

1. The services which are to be achieved by the application.
2. The source and format of information which will be used to support the application.
3. The short and long-term plans of the user with regard to the application.
4. Interfaces to other applications existing or planned.

Phase I deliverables are:

1. A subsystem specification, in IDD input form, stored on a designated PSL, containing the information described in Sections 1, 2, and 3 of the subsystem specification standard described in Appendix D of the

Design Guide. Four copies of the subsystem specification will also be supplied in document form.

2. A service analysis description for each service function to be supported by the application prepared in accordance with Appendix A.10 of the Design Guide. Output services will include additional description information described in Appendix A.11. Four copies of the service analysis will be supplied in document form.

Phase I Begin Date -

Delivery Date -

Two working weeks will be allowed for NIPSSA review of the deliverables from the time of receipt at NIPSSA.

Scheduled Phase I Completion Date -

## 2. PHASE II. DATABASE FOUNDATION DESIGN.

This phase of the database design is concerned with the definition of data elements, element groups, and element group entity relationships. Each step of this phase must be performed carefully and methodically to insure an effective design.

Of the five steps in Phase II, the contractor performs three. The remaining two are performed by the DA staff.

While deliverables are specified at the end of each contractor-performed step of Phase II, a single formal deliverable will be supplied by the contractor at the completion of the Phase. This deliverable will constitute the combined deliverables of steps 2, 4, and 6. It will be delivered in the form of a listing of data elements, groups, and entities extracted from IDD card-image entries stored on a NIPSSA-supplied program support library (PSL). Steps 2 and 4 deliverables will be supplied in IDD card-image form on hard copy output.

## 2.1. Step 2. Initial Data Element Definition.

Step 2, the initial data element definition, is performed by the contractor. Appendix A.6 of the design guide provides the guidelines for this step.

Deliverables for this step are:

1. A complete description of each elementary data element which is required to support the requested application. Each element will be described using Appendix A.6 of the Design Guide as the preparation guideline.
2. The data element descriptions will be stored in IDD image form on a PSL designated by the NIPSSA project manager. Four copies of the element descriptions in document form will be supplied to the NIPSSA project manager.

Step 2 Begin Date -

Delivery Date -

## 2.2. Step 3. DA Staff Review of Data Element Definitions.

Step 3, the DA staff review, is performed by NIPSSA personnel. Contractor analyst personnel should be available for consultation when questions arise. Ten working days are allowed for performance of this step.

When the DA review is complete, the project manager prepares a summary identifying the errors or potential problem areas. This summary is presented to the contractor for correction during step 4.

Step 3 Scheduled Begin Date -

Scheduled Delivery Date -

## 2.3. Step 4. Data Element Group Definition.

Step 4, group elements, is performed by contractor



analysts following correction of errors and discrepancies found during the DA review.

Appendix A.7 describes the procedure for preparing IDD entries for data element groups. Once the IDD entries have been prepared, they should be loaded to the PSL after the last elementary data element entry.

Deliverables for step 4 are:

1. Stored PSL entries correcting errors and discrepancies found during the DA review (step 3). The hard copy element list is updated by the contractor and report listing produced for review.
2. IDD entries, stored on a PSL designated by the project manager, defining the initial grouping of data elements supporting the application. Four copies of the group element definitions will be provided to the project manager in document form. Appendix A.7 of the Design Guide is the guideline.

Step 4 Begin Date -

Delivery Date -

#### 2.4. Step 5. DA Review of Group Element Definitions.

Step 5, group element review, is performed by the DA staff. The process described for step 3 above is repeated. Ten working days are allowed for this step.

The project manager prepares a summary of discrepancies as in step 3 and reviews the discrepancies with the contractor. One working week is provided for NIPSSA to perform the review after receipt of the step 4 deliverables.

Scheduled Step 5 Begin Date -

Scheduled Step 5 Completion Date -



## 2.5. Step 6. Data Entity Relationship Definition.

Step 6, data entity relationships, is performed by the contractor. This step is the last one in the definition process. The elements and element groups defined previously are associated into logical data entities. These entities will appear as record types to the end user.

Deliverables for this step are:

1. Corrections to element and group definitions found as a result of the DA review (step 5). The corrections will be entered into the PSL and a report listing prepared.
2. Definitions of group entities which represent the final association of elements before database integration. Section 3.2 of the Design Guide and Appendix A.7 are the guidelines for preparation of this data. The definitions will be stored on the PSL following the last group definition of step 4. Four copies of the group entity definitions will be provided to the project manager in document form.

Step 6 Begin Date -

Delivery Date -

## 3. PHASE III. DATABASE INTEGRATION

During Phase III, the definitions of Phase II are integrated into the database structure. Two task steps accomplish the integration. Both are performed by the DA staff.

Three working weeks are allowed for the completion of Phase III. This may be adjusted where an application is larger or smaller than the 100 element average.

At the completion of Phase III, a schema will be ready for development use. DMCL and subschema recompilation

will be complete and the dictionary ready for development of supporting software.

Scheduled Phase III Begin Date -

Scheduled Phase III Completion Date -

#### 4. PHASE IV. BATCH INPUT PROCESSING AND FILE CONVERSION DEVELOPMENT.

This phase of the application project begins actual development of software to support the defined database. Software developed during Phase IV will be used to load data to the database through batch processing. It is assumed that data will be presented to the system in card-image form whether prepared on punched cards, magnetic tape, or disk.

Where an existing automated or machine-readable source of information will be converted and loaded to the database, it is necessary that conversion software be prepared in addition to writing batch IP's. Most conversion requirements can be effectively met through use of CULPRIT as the conversion processor.

The functions of this phase are performed by contractor analysts. Regular consultation with the project manager is encouraged.

##### 4.1. Step 9. Definition of Batch Input Processing Formats.

Step 9 of the application development defines the input processing formats.

Once the physical layout of the IP has been defined, the layout is described in IDD form as described in Appendix A.12. These descriptions are prepared in card image form and delivered to the project manager for loading into the data dictionary.

Deliverables of this step are:

1. Layout definitions of all IP formats to be generated to support store, modify, and delete of data. Each IP will be defined as described in Appendix A.12. Pictorial

coding sheets for each IP will be prepared using those in Appendix F of the design guide as examples.

2. Layout definitions of all IP's required to associate defined database entities. Each will be defined by pictorial coding sheets using Appendix F of the Design Guide as examples.

Step 9 Begin Date -

Delivery Date -

#### 4.2. Step 10. Preparation of Batch IP's.

Step 10 is the preparation of IP parameter statements for each defined IP. This step is performed by contractor analysts. This step may be performed in parallel with step 9 above.

The deliverables for step 10 are:

1. IP parameters for all IP's used for store, modify, and delete functions. IP parameters will be prepared as described in Appendix A.12 of the Design Guide and stored on PSL as separate members. Every database entity created or modified for the applications will have at least a store and a delete IP prepared. One or more modify IP's will be defined where appropriate.
2. Each defined IP will be compiled using program generation software supplied by NIPSSA. Each IP will result in a cataloged object program stored on a linkage library defined by the NIPSSA project manager.

Step 10 Begin Date -

Delivery Date -

#### 4.3. Step 11. Batch Test Data Preparation.

Step 11 is the most time-consuming step of this Phase. Comprehensive test data must be prepared for each IP. Test data must:

1. Test each IP for all possible DATABASE conditions which can affect IP accuracy.
2. Test each IP for all possible DATA CONTENT conditions which can affect IP accuracy.

Prepared test data will be stored on the PSL designated by the project manager, listed and provided to the project manager for review.

Deliverables for this step are:

1. Test data prepared for each IP. The test data will be stored in separate PSL members for each IP or logical IP group.
2. A list of test data accompanied by a description of the test approach and the anticipated results of each test.

Step 11 Begin Date -

Delivery Date -

#### 4.4. Step 12. Batch IP Testing.

Each IP is thoroughly tested in step 12. All test data is applied and error conditions reviewed. Processing errors are corrected and tests rerun. Test results are reviewed by the project manager.

Deliverables for step 12 are:

1. Results of tests run against the IP's. A comparison of the test results against the results expected. Where the results differ, IP's will be corrected and retested until correct results are achieved.
2. Corrected IP definitions, object programs,

and test results against the corrected IP's.

Step 12 Begin Date -

Delivery Date -

#### 4.5. Step 13. Initial Subsystem Users Guide Preparation.

The initial subsystem users guide is prepared in step 13. Sections 1 and 2 of the users guide are prepared similar to those shown in Appendix F. Individual coding sheets and instructions are prepared for each IP generated. Additional coding sheets and instructions are prepared for each association (connect and disconnect) function required using standard association IP's.

Batch input coding instructions will be grouped for logical use. For example, those instructions for the initial storing of a database record occurrence are grouped together as are those to modify the database. Frequently used coding instructions are placed in a group at the beginning of the section.

Separator tabs aligned to groups of functions will be provided to make locating specific coding instructions easier. The page location of each instruction will be provided in the table of contents.

Deliverables for step 13 are:

1. Sections one and two of the batch users guide for the application. The format and general content illustrated in Appendix F of the Design Guide will be used. The content will be tailored to the application.
2. Section three of the batch users guide for the application. The format illustrated in Appendix F of the Design Guide will be followed. Each IP module created in support of the application will be represented by a coding sheet and instructions. The instructions will explain a sample completed coding sheet.
3. Each coding sheet will be identified in the



table of contents. Coding instructions will be grouped by logical functions to make use easier. Each function will be separated by index tabs.

Step 13 Begin Date -

Delivery Date -

#### 4.6. Step 14. Batch System Operational Test.

Step 14 is the operational test of the batch system capability. The user personnel who will utilize the system are trained and assisted in initial use of the system. A training class of one day duration (maximum) will be held for each of these user areas:

1. User management. This class will describe the system capabilities from a management viewpoint. The part the system will play in improved management of security will be emphasized. Visual aids and handouts will be prepared to acquaint management with the system and its interface with other facets of NICOLS.
2. User personnel who will operate the system. This class will describe the operation of the system in more detail. The users guide will be thoroughly reviewed. These points will be covered:
  - a. JCL and job submission.
  - b. Data entry preparation.
  - c. Coding conventions where they exist.
  - d. Report preparation.

Visuals aids and handouts will be prepared to support the training.

Deliverables for step 14 are:

1. Visual aids and handouts for classes training management and user data entry personnel in the use of the application.
2. One one-day (maximum) course to acquaint user management with the system and its capability to support the organization.
3. Two one-day (maximum) courses for user data entry and functional personnel who will directly interface with the application.
4. Two person-weeks of direct user support.

Step 14 Begin Date -

Delivery Date -

#### 4.7. Step 15. Define Conversion Criteria.

Step 15 defines conversion criteria for each of the existing machine-readable files currently in use. This information includes:

1. The source file and location of each data element.
2. The object IP format and the position of each data element.
3. Differences in length and mode of each data element, where occurring, between source file and database.
4. Conversion requirements, where required, of each data element. This includes translation of codes, special verification of source data, and optional purging of data.

Deliverables for step 15 are:

1. Conversion definitions for each data element to be transferred from machine-readable

existing files to the database.

2. Conversion sequence when more than one file is to be converted.

Step 15 Begin Date -

Delivery Date -

#### 4.8. Step 16. Conversion Software Development.

The conversion software is prepared in step 16. Using the conversion criteria prepared in the previous step, CULPRIT modules are prepared to effect the conversion of existing machine-readable files into IP format.

The files are converted and the output listed prior to loading of data. Listings are reviewed to insure conversion criteria have been met. Upon approval by the project manager, the converted files are loaded to the operational database.

Deliverables of step 16 are:

1. Software programs and/or CULPRIT modules which will convert all machine-readable files associated with the application to input processing format for loading of the database.
2. Instructions and JCL decks required to perform the conversion.
3. Listing of dummy conversion run showing the conversion process, errors in data encountered, and the IP data created.
4. Upon approval by the project manager and correction of any errors, the actual conversion is performed. The results of conversion and database loading are provided to the project manager for review and approval.

Step 16 Begin Date -

Delivery Date -

NOTE: All interim deliverables described during Phase IV are formally deliverable on the above date. The products delivered on the above date, therefore, represent the end product of batch input processing software development, testing documentation, and conversion software. Training and operational test deliverables are considered to be a joint deliverable of Phases IV and V. IT IS A REQUIREMENT THAT PHASE V BE COMPLETED BEFORE PHASE IV (OPERATIONAL TEST STEP) CAN BE COMPLETED.

PHASE V. BATCH OUTPUT SUPPORT DEFINITION AND DEVELOPMENT.

This phase of the application development is accomplished in three steps, all performed by the contractor:

1. Step 17 expands on the service analysis performed during step 1 and defines the output services in detail. This includes identifying the format of outputs and the database resources required to satisfy the output service.
2. Step 18 converts the specifications of step 17 into programmed output capabilities. CULPRIT and, where required, COBOL report writer, are used to produce and test each output service.
3. Step 19 provides updates to the users guide defining each output service, illustrating the output, and describing the procedures for obtaining the output.

5.1. Step 17. Service Analysis Expansion for Outputs.

Step 17 expands upon the service analysis information placed in the data dictionary. The physical layout of the report is prepared.

AD-A069 868

NAVAL INTELLIGENCE PROCESSING SYSTEMS SUPPORT ACTIVIT--ETC F/6 9/2  
INTEGRATED DATABASE DEVELOPMENT AND DESIGN GUIDE. VERSION 2.0.(U)  
MAY 79

UNCLASSIFIED

NIPSSA-UM-8000/2.0-5/79

NL

7 OF 7

AD  
A069868



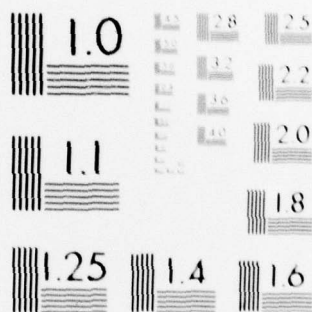
END

DATE  
FILMED

7-79

DDC





MICROCOPY RESOLUTION TEST CHART  
NATIONAL BUREAU OF STANDARDS 1963-A

A comprehensive narrative description of the output will be added to the service analysis description and stored in the data dictionary. Complete pictorial presentations of printed or displayed reports, cover pages, and associated displays will be prepared in sufficient detail that development of the capabilities in the following step will require little, if any, additional analysis.

Deliverables for step 17 are:

1. Updated narrative description of each batch output service store in the data dictionary.
2. Where output services are printed reports, an annotated pictorial of the report.
3. Where output services are other than printed reports, a detailed pictorial definition of the format of the output service.

Step 17 Begin Date -

Delivery Date -

#### 5.2. Step 18. Batch Output Preparation.

Step 18 uses the comprehensive analysis and definition of the output capabilities defined in step 17 to produce software necessary to satisfy the desired output service. CULPRIT will be used wherever possible for preparing output facilities to be executed in a batch environment. COBOL Report Writer may be used with the approval of the project manager.

Each output capability created will be thoroughly tested. Test results will be reviewed by the project manager and compared against the specifications of the requested service.

Deliverables for step 18 are:

1. Executable output capabilities prepared using either CULPRIT or (with the project manager's approval) COBOL Report Writer.

2. All source statements or parameters for output service capabilities will be stored on a designated PSL.
3. Test results for each output service produced. Where the output service is other than a printed report, a printed display of the output contents which can be used to verify the output accuracy.

Step 18 Begin Date -

Delivery Date -

### 5.3. Step 19. Batch Users Guide Update for Outputs.

Step 19 provides the user with instructions and examples of the utilization of the output service. Each output service will be individually documented and placed in the users guide. Each entry will contain:

1. Instructions for requesting the output service, including a description of the JCL and control cards required. Control card fields will be described in detail. Illustrations of typical control card configurations will be included.
2. Definitions of options associated with the output service, if any. Where options are available, each will be described separately and illustrated.
3. An illustration of the normal output of the service where the output is visually presented.

Deliverables for this step are:

1. Users guide entries for each output service prepared under the previous step. Users guide entries will contain the information described above.
2. The users guide table of contents will be updated to reflect the additions. Index

tabs will be added to improve usability of the guide.

Step 19 Begin Date -

Phase V Begin Date -

Phase V Delivery Date -

NOTE: The formal deliverables of Phase V are the composite informal deliverables of each step within the Phase.

At the completion of this phase, a complete batch processing capability of the application is available to the user.

#### 6. PHASE VI. ON-LINE INPUT PROCESSING.

On-line input processing provides application users with the ability to perform data entry and correction from an interactive CRT. This phase is similar to Phase IV except that data conversion and initial users guide preparation have already been performed.

Phase VI is composed of five steps, all performed by contractor personnel:

1. Step 20 defines the format of on-line input processing screens after having determined what data entry and corrections requirements are present for the application. Not all application data may be suited for on-line input processing.
2. Step 21 prepares on-line input processing program parameters based on the specifications developed in step 20. The parameters are reviewed and stored on PSL in preparation for compilation.
3. Step 22 utilizes the parameters developed in the previous step and creates on-line input processing programs for data entry. Each program is thoroughly tested using test data developed in this step. Corrections are made as required and the PSL updated. Test



data descriptions are stored on a designated PSL for future use. The project manager reviews test results against the specifications to insure correct processing.

4. Step 23 updates the users guide to include an instruction for each on-line input processing screen. Instructions will identify each data element used, all processing options, function key usage, and illustrate each screen.
5. Step 27 is normally performed after completion of Phase VII (on-line output services). This step includes the training of user operating personnel, development of training materials, and direct user assistance for a short period of time.

#### 6.1. Step 20. On-line Input Processing Definition.

Step 20 is the on-line input processing definition step. Service analysis information is reviewed and requirements for on-line data entry and correction/modification services are translated into specifications for teleprocessing screens. The physical layout for the screens is prepared and reviewed. Descriptions of screen functions are prepared and added to the service analysis description. The logical progression from one screen to another where screen functions are related is defined and function keys assigned as required.

##### Deliverables are:

1. Specifications for teleprocessing screens updating the service analysis in the data dictionary.
2. Pictorial layouts of display screens, identifying each field and function.
3. Logical progression pictorial(s) describing logical follow-on screens and the function key assignments.

Step 20 Start Date -



Delivery Date -

6.2. Step 21. On-line IP Parameter Definition.

Step 21 utilizes the descriptions prepared in the previous step to generate on-line IP parameters and data dictionary entries. Appendix A.14 provides detailed instructions for preparation of screen parameters.

Deliverables are:

1. On-line IP screen descriptions stored in the data dictionary in accordance with Appendix A.14 of the Design Guide.

Step 21 Start Date -

Delivery Date -

6.3. Step 22. On-line IP Software Generation and Test.

Step 22 compiles the screens from the parameters stored on PSL and in the data dictionary. Each screen is tested using a set of test data developed for the purpose. Test data will be developed using the same criteria as in step 11 (batch test data preparation). Test data descriptions will be stored on PSL in narrative form for future use. Test results will be reviewed by the project manager to insure that all requirements have been satisfied.

Deliverables are:

1. Compiled and tested on-line IP modules.
2. Narrative description of test data stored on PSL.
3. Demonstrable tests results for approval by the project manager.

Step 22 Start Date -

Delivery Date -

#### 6.4. Step 23. On-line Input Processing Users Guide.

Step 23 is the preparation of user guide information for on-line input processing. An entry for each screen will be included which:

1. Illustrates the screen.
2. Describes each data field to be entered on the screen.
3. Defines the association of other screens which are logically appended to a screen. This association description will identify the function keys or command action required to reach associated screens. A pictorial diagram of screen relationship will be included.

Deliverables for step 23 are:

1. A user guide entry for each on-line IP prepared in the previous step. The entry will contain the information described above.
2. An update to the table of contents of the users guide to provide ready reference to screens. Index tabs will be used and entries organized to achieve maximum utilization of the guide.

Step 23 Start Date -

Delivery Date -

#### 6.5. Step 27. On-line System Operational Test.

Step 27 is the final step of on-line system implementation. Phase VII must be complete, or omitted, prior to performing this step. The step is the operational test of software and training of user personnel. A one-day (maximum) training class for user operating personnel is held to acquaint users with the capabilities of the system. Training materials and handouts are prepared to support the class and are turned over to the project manager for use in

future training. The operation of the system is observed and discrepancies noted. Corrections are made where necessary and the system documentation and programs updated. The project manager reviews the system operation and approves the system for turnover.

Deliverables for step 27 are:

1. Training materials and handouts for teaching user operating personnel in the utilization of the system.
2. Two one-day (maximum) training classes for user operating personnel to familiarize them with the operation of the system.
3. Corrections and updates to documentation and programs, as required, to correct malfunctions in contractor produced software or software parameters.
4. Two person-weeks of direct support to user operating personnel.

Step 27 Start Date -

Delivery Date -

NOTE: All informal deliverables defined as part of Phase VI are due on the above date. PHASE VI (STEP 27) CANNOT BE COMPLETED UNTIL PHASE VII HAS BEEN COMPLETED.

#### 7. PHASE VII. ON-LINE OUTPUT DEFINITION AND DEVELOPMENT.

This phase is nearly identical to Phase V except that on-line display output is being developed.

Phase VII steps are:

1. Step 24 expands on the service analysis definitions of on-line output services. Each requested service is carefully analyzed to determine which of the four development modes described above will be used to implement the service. Detailed descriptions of the services are added to

the service analysis prepared in Phase I and the data dictionary updated.

2. Step 25 is the preparation of output programs or queries to satisfy the service request. Each program or query is stored on a designated PSL. Each capability is tested and test results reviewed by the project manager.
3. Step 26 is the preparation of updates to the users guide reflecting each of the on-line services provided.

#### 7.1. Step 24. On-line Output Service Analysis Update.

Step 24 requires a thorough review of those service requests prepared during Phase I. Each request which requires on-line output support is evaluated. One of the development approaches defined above is selected for each service requested. The detail definition of the visual display to be produced is prepared. Specifications are expanded to include complete details about the service, its purpose, and the depth of logic required to produce the capability. The data dictionary is updated with the additional information. CAUTION: When updating remarks of the service analysis, first update the information on the PSL and then update the data dictionary from the PSL.

Deliverables are:

1. Expanded specifications from the original service analysis defining the development approach for each output service.
2. A pictorial of each proposed output service, annotated to describe each field and function.
3. The data dictionary updated with expanded service analysis information and specifications.

Step 24 Start Date -

Delivery Date -



## 7.2. Step 25. On-line Output Software Development.

The actual software or query modules are prepared and tested during Step 25. Each output service is thoroughly tested and the results approved by the project manager.

Deliverables are:

1. Software module parameters, for those output services created using CULPRIT. The module parameters will be stored on a PSL designated by the project manager.
2. Software source programs, for those output services created using COBOL. The source programs will be stored on a PSL designated by the project manager. Source programs will be documented internally and structured using structured programming techniques and conventions described in Appendix B.7 of the Design Guide.
3. Query parameters, for those output services satisfied through use of ON-LINE QUERY, stored in the data dictionary.
4. Test results demonstrated to the project manager to insure that all output services meet the defined specifications.

Step 25 Start Date -

Delivery Date -

## 7.3. Step 26. On-line Users Guide Reports Update.

Step 26 is the updating of the users guide to reflect the instructions for use of each service request. The users guide will contain, for each output service:

1. A pictorial of the output service.
2. Complete step-by-step instructions for requesting the output from a CRT terminal.



3. A description of any options available with the output service. Where the option changes the format or function of the service, an illustration and explanation of each option will be included.

Deliverables for step 26 are:

1. Entries for each on-line output service prepared during the previous step. Each entry will contain the information described above.
2. An update to the table of contents of the users guide to reflect the addition of the on-line output services. Index tabs will be inserted and output service descriptions grouped to achieve the most effective use of the users guide.
3. The reproducible master copy of the complete application users guide and 10 copies. Each copy will be punched and enclosed within a 3-ring standard binder. The users guide will be formatted and page numbered as illustrated in Appendix F of the Design Guide. The document number will be supplied by the project manager.

Step 26 Start Date -

Delivery Date -

NOTE: All informal deliverables defined within Phase VII are deliverable formally on the above date.